AN EFFICIENT DATA ENCRYPTION MECHANISM BY USING OPTIMAL LSB TECHNIQUE

A Project Report submitted in the partial fulfilment of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY
In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Shaik Thameem Ansari (19471A0558)

Kandregula Chaitanya (19571A0527)

N.S.V.Bala Krishna (19471A0540)

Under the esteemed guidance of

Dr.S. Siva Nageswara Rao, M.Tech., Ph.D

Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Cycle -1
NIRF rank in the band of 251-320 and an ISO 9001:2015 Certified
Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada
KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2022-2023

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "An Efficient Data Encryption Mechanism By Using Optimal LSB Technique" is a bonafide work done by the team Shaik Thameem Ansari (19471A0558), Kandregula Chaitanya (19571A0527), N.S.V.Bala KRISHNA (19471A0540) in partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2022-2023.

PROJECT GUIDE

Dr.S. Siva Nageswara Rao, M.Tech., Ph.D.,

Professor

PROJECT CO-ORDINATOR

Dr.M.Sireesha ,MTech.,Ph.D.,

Assoc. Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,

Professor

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M.V.Koteswara Rao,** **B.Sc.**, who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr.M.Sreenivasa Kumar**, **M.Tech., Ph.D., MISTE., FIE(I).**, for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr.S.N.Tirumala Rao,** **M.Tech.,Ph.D.**, HOD of CSE department and also to our guide **Dr.S.Siva Nageswara Rao,** **M.Tech.,Ph.D.**, Professor of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Dr.M. Sireesha,** **M.Tech.,Ph.D.**, Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree. We have no words to acknowledge the warm affection, constant inspiration, and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

Shaik Thameem Ansari (19471A0558)
Kandregula Chaitanya (19471A0527)
N.S.V. Bala Krishna (19471A0540)

DECLARATION

We declare that this project work titled “AN EFFICIENT DATA ENCRYPTION MECHANISM BY USING OPTIMAL LSB TECHNIQUE” is composed by ourself that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

By

Shaik Thameem Ansari(19471A0558)

Kandregula Chaitanya(19471A0527)

N.S.V.Bala Krishna(19471A0540)

ABSTRACT

In today's world the art of sending & displaying the hidden information especially in public places, has received more attention and faced many challenges. Therefore, different methods have been proposed so far for hiding information in different cover media. It is well known that encryption provides secure channels for communicating entities. However, due to lack of covertness on these channels, an eavesdropper can identify encrypted streams through statistical tests and capture them for further cryptanalysis.

Information hiding is an emerging research area, which encompasses applications such as copyright protection for digital media, watermarking, fingerprinting, and steganography. In watermarking applications, the message contains information such as owner identification and a digital time stamp, which usually applied for copyright protection. Fingerprint, the owner of the data set embeds a serial number that uniquely identifies the user of the data set. This adds to copyright information to makes it possible to trace any unauthorized used of the data set back to the user.

NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering andresearch

M2: Build a passionate and a determined team of faculty with student centric teaching,imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values instudents for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools andtechnologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and cancommunicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineeringproblems.

2. Problem analysis: Identify, formulate, research literature, and analyzecomplex engineering problems reaching substantiated conclusions using firstprinciples of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complexengineering problems and design system components or processes that meetthe specified needs with appropriate consideration for the public health andsafety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-basedknowledge and research methods including design of experiments, analysis andinterpretation of data, and synthesis of the information to provide validconclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques,resources, and modern engineering and IT tools including prediction andmodeling to complex engineering activities with an understanding of thelimitations.

6. The engineer and society: Apply reasoning informed by the contextualknowledge to assess societal, health, safety, legal and cultural issues and theconsequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of theprofessional engineering solutions in societal and environmental contexts,and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leaderin diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receiveclear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engagein independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO'S):

CO425.1: Analyse the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements.

CO425.3: Review the Related Literature **CO425.4:**

Design and Modularize the project

CO425.5: Construct, Integrate, Test and Implement the Project.

CO425.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1		√											√		
C425.2	√		√		√								√		
C425.3				√		√	√	√					√		
C425.4			√			√	√	√					√	√	
C425.5					√	√	√	√	√	√	√	√	√	√	√
C425.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1	2	3											2		
C425.2			2		3								2		
C425.3				2		2	3	3					2		
C425.4			2			1	1	2					3	2	
C425.5					3	3	3	2	3	2	2	1	3	2	1
C425.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. **Low level**

2. **Medium level**

3. **High level**

Project mapping with various courses of Curriculum with AttainedPO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C3.2.4, C3.2.5	Gathering the requirements and defining the problem, plan to develop a**n efficient automated attendance using deep learning.**	PO1, PO3
CC4.2.5	Each and every requirement is critically analyzed, the process model is identified and divided one module.	PO2, PO3
CC4.2.5	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC4.2.5	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC4.2.5	Documentation is done by all our three members in the form of a group	PO10
CC4.2.5	Each and every phase of the work in group is presented periodically	PO10, PO11
CC4.2.5	Implementation is done and the project will be handled by the college management and in future updates in our project can be done based on project done.	PO4, PO7
CC4.2.8 CC4.2.	The project includes hardware components like camera,& connector.	PO5, PO6

INDEX

LIST
OF
FIGURES

CHAPTER - 1

INTRODUCTION

The word steganography means" covered in hidden writing". The object of steganography is to send a message through some innocuous carrier (to a receiver while preventing anyone else from knowing that a message is being sent to all. Computer based steganography allows changes to be made to what are known as digital carriers such as images or sounds.

In steganography information can be hidden in carriers such as images, audio files, text files, and video and data transmissions. When message is hidden in the carrier a stego carrier is formed for example a stego-image. Hopefully it will be perceived to be as close as possible to the original carrier or cover image by the human senses.

Images are the most widespread carrier medium. The are used for steganography in the following way. The message may firstly be encrypted. They are used for steganography in the following way. The message may firstly be encrypted. The sender embeds the secret message to be sent into a graphic file. This results in the production of what is called stego-image. Additional secret data may be needed in the hiding process e.g. a stegokey etc. This stego-image is then transmitted to the recipient.

This project report intends to give an overview of image steganography, its uses and techniques. It also attempts to identify the requirements of a good steganography algorithm and briefly reflects on which steganographic techniques are more suitable for which applications. Steganography become more important as more people join the cyberspace revolution. Steganography is the art of concealing information in ways that prevents the detection of hidden messages. Steganography include an array of secret communication methods that hide the message from being seen or discovered.

1.1 GOALS AND OBJECTIVES

To develop an encryption system that can provide security through obscurity to solve the problems mentioned above.

1.1.1 MAIN OBJECTIVE

The main objective of this project is to develop an encryption system using image steganographic method, which will help the users encrypt the information and data in storage and transmission on a network.

1.1.2 SPECIFIC OBJECTIVES

The project seeks to achieve the following:

- To provide a secret form of encryption through obscurity using image steganography.
- To make all confidential information on both their systems and client systems encrypted to prevent unauthorized access, viewing, use, suspicion and modification.
- To provide a steganographic encryption platform for confidential communication and secret information sharing and storing.
- The encrypted image must avoid drawing suspicion to the existence of hidden information in it.

1.2 USE OF PROPOSED SYSTEM

The user needs to run the steganographic application. The user has two options tabs; Encrypt and Decrypt. If user select encrypt tab, application give the user a screen to select any image file (carrier or cover file) and also an information file, then an option to save the encrypted stego-image to a destination folder.

If user selects decrypt tab, application gives the user an option to select only a bitmap image file (stego - image) and a path where the user wants to save the hidden extracted information file to.
This project has two methods – encrypt and decrypt.

i. In Encryption the information file is embedded into an image file (carrier or cover file).
ii. In Decryption is getting the information from the encrypted image (stego-image).

1.3 DEFNITION OF TERMS

This section of project includes important or key terms that should be substantially and clearly defined according to how they are used in the study in order to facilitate understanding of the problem and avoid ambiguous meaning to terms, which can be otherwise interpreted in different ways.
Definitions of terms may be of two categories:

- Conceptual Definition.
- Operational Definition.

(A). Conceptual Definition:

These are words usually taken from the dictionary. It carries a universal meaning easily understood by people.

- Cipher: A cryptographic system in which units of plaintext of regular length, usually letters are arbitrarily transposed or substituted according to a predetermined code.
- Cypher text: This is an encrypted text. Plaintext is what you have before encryption. Cipher text is the encrypted result.
- Encryption: The processes of coding or scrambling information in a way that it can only be decoded and read by someone who has the correct decoding key or algorithm. (Obscures information)
- Decryption: The process of decoding data that has been encrypted into a secret format with a known key or algorithm.(recovers the information)
- Algorithm: This is a step-by-step procedure or formula for solving a problem or accomplishing some tasks especially by a computer.
- Information file: This is a file that contains accurate, up-to-date documents.

(B). Operational Definition:

These words express the meaning of the terms as used in a particular field of study

- Cover object / carrier file: A file which is used to hide information inside of it.
- Steganalysis: The process of detecting hidden information inside of a stego-file.
- Stego-file/ stego-image: The file in which the information is hidden.
- Least significant bits: Pieces of information inside a file which can be overwritten or altered without damaging the file.

1.4 SYSTEM REQUIREMENTS

1.4.1 Hardware Requirements:

- System Type : Intel Core i3 or above
- Cache Memory : 4MB (Megabyte)
- RAM : 8 gigabytes (GB)
- A minimum video graphic acceleration of 256MB RAM
- A minimum free hard disk space of 1 TB.

1.4.2 Software Requirements:

- Operating System : Windows 10 Home, 64-bit Operating System.
- Coding Language : Java, Html & CSS
- IDE : Visual Studio Code
- Browser : Any latest browser like chrome

CHAPTER - 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Due to advances in Information Communication Technology (ICT), most of information is kept electronically. The role of information technology (IT) as an integral part of our daily life makes information security critical for individuals and organizations. Consequently, the security of information has become a fundamental issue. The amount of personal and corporate information transmitted and stored on networks, and the variety of threats to that information, combine to form a pressing need for increased protection of that information. It is of no surprise that countless encryption methods like cryptography, steganography, fingerprint and watermarking are used by organizations and individuals to protect such information.

Besides cryptography, steganography can be employed to secure information. Steganography is a technique of hiding information in digital media. In contrast to cryptography, the information or encrypted message is embedded in a digital host before passing it through a network or storing it, thus the existence of the information is unknown. Besides hiding data for confidentiality, this approach of information hiding can be extended to copyright protection for digital media like audio, video, and images.

The growing possibilities of modern communications require the use of secure means of protecting information. The most common method of protecting information is cryptography whereby the information is scrambled into unintelligible stream that cannot be decrypted by the casual viewer. Steganography is an approach in information hiding whereby the information is hidden inconspicuously inside a host data set such that its presence is imperceptible.

Basically, the purpose of cryptography and steganography is to provide secret communication. However, steganography must not be confused with cryptography. Cryptography hides the contents of the secret information from malicious people, whereas steganography conceals the very existence of the information. Therefore, the methods used in breaking the system are different.

In cryptography, the system is broken when the attacker can decrypt the unreadable data to form back the secret information. But to extract hidden information that is embedded using steganography, the attacker first of all need to realize the very existence of the secret information. Without this knowledge, the secret data can pass through even right under his or her nose. Also in cryptography, the structure of a n information is scrambled to make it meaningless and unintelligible unless the decryption key or algorithm is available. It makes no attempt to disguise or hide the encoded information.

Basically, cryptography offers the ability of storing and transmitting information between persons in a way that prevents a third party from reading it. Cryptography can also provide authentication for verifying the identity of someone or something. In contrast, steganography does not alter the structure of the secret information, but instead hides it inside a cover-image so that it cannot be seen or known.

A message in a cipher text (encrypted message), for instance, might arouse suspicion on the part of the recipient while an “invisible” message created with steganographic methods will not. In other words, steganography prevents an unintended recipient from suspecting that a secret message exists. In addition, the security of classical steganography system relies on secrecy of the data encoding system.

Once the encoding system is known, the steganography system is defeated. It is possible to combine the techniques by encrypting message using cryptography and then hide the encrypted message using steganography. The resulting stego-image (encrypted image) can be transmitted or stored without revealing that secret information is being exchanged or preserved.

Furthermore, even if an attacker were to defeat the steganographic technique and suspect the message from the stego-object (encrypted image), he would still require the steganographic decoding algorithm (steganographic system) to decipher the encrypted message. Common techniques used in steganography are least significant bit insertion (LSB), masking and filtering, and transformation techniques.

In this project I present least significant bit insertion techniques (LSB), which randomly select the least significant pixels of the cover-object that is used to hide the Information file.

2.2 OVERVIEW OF STEGANOGRAPHY

The word steganography comes from the Greek word “Steganos”, which means covered or secret and “graphy” which means writing or drawing. Therefore, steganography means, covered writing. Steganography is the art and science of hiding information such that its presence be detected. Secret information is encoded in a way such that the very existence of the information is concealed in a human perceptible.

Steganography is an ancient technology that has applications even in today’s modern society. Steganography has taken many forms since its origin in ancient Greece. The first recorded use of the term can be traced back to 440 BC. During the war between Sparta and Xerxes. Dermeratus wanted to warn Sparta of Xerxes’ pending invasion. To do this, he scraped the wax off one of the wooden tablets they used to send messages and carved a message on the underlying wood. Covering it with wax again, the tablet appeared to be unused and thereby slipped past the sentries’ inspection. Herodotus, who documented the conflict between Persia and Greece in the fifth century B.C., felt that the art of secret writing saved Greece from Xerxes, the tyrant king of Persia.

However, this would not be the last time steganography would be used in times of war. In World War II, the Germans utilized this technology. Unlike the Greeks, these messages were not physically hidden; rather they used a method termed “null ciphering.” Null ciphering is a process of encoding a message in plain sight. For example, the second letter of each word in an innocent message could be extracted to reveal a hidden message.

A message sent by a German spy during World War II read: **“Apparently neutral’s protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects for pretext embargo on by-products, ejecting suits and vegetable oils”**.

By taking the second letter of every word, the hidden message is **“Pershing sails for NY June 1”** can be retrieved [12]. Also, during the American Revolution, invisible ink which would glow over a flame was used by both the British and Americans to communicate secretly [13].

More recent cases of steganography include using special inks to write hidden messages on bank notes and also the entertainment industry using digital watermarking and

fingerprinting of audio and video for copyright protection. Two other technologies that are closely related to steganography and cryptography are water marking and fingerprinting. These technologies are mainly concerned with the protection of intellectual property; thus, the algorithms have different requirements than steganography and cryptography.

The main goal of steganography is to communicate securely in a completely undetectable manner and to avoid drawing suspicion to the transmission of a hidden data [16]. Therefore, in existing communication methods, steganography can be used to carry out hidden exchanges. The idea of steganography is to keep others from thinking that the information even exists and not to keep others from knowing the hidden information.

If a steganography method causes anybody to suspect there is secret information in a carrier medium, then the method has failed [17].

2.3 DIFFERENT KINDS OF STEGANOGRAPHY

There are several suitable media that can be used as cover-objects such as image, audio and video. Message is the data that the sender wishes to keep confidential and will be embedded into the cover-object by using a stego system encoder. There are several suitable carriers below to be the cover-object:

2.3.1 Network Protocols such as TCP, IP and UDP.

2.3.2 Audio that is using digital audio formats such as wav, midi, mp3 and voc.

2.3.3 Video files such as mpeg, mp4, flv.

2.3.4 Images file such as bmp, gif and jpg.

A stegosystem encoder can be represented by using the following relation: Z = *f*

(C, M, K)..... (1)

Where Z' is the stego-object C is the cover-object

M is the message or information file K is the stego-key. (Optional)

A stego-key which is optional is a password, which ensures that only the recipient who knows the corresponding decoding key will be able to extract the message from a cover- object. The output of the stegosystem encoder is known as the stego-object.

2.4 RECOVERING INFORMATION FILE FROM A STEGO-OBJECT

Recovering message from a stego-object requires the cover-object itself and a corresponding decoding key if a stego-key was used during the encoding process. The original image may or may not be required in most applications to extract the message. In general, the information hiding process extracts redundant bits from cover- object. The process consists of two steps,

i. Identification of redundant (least significant) bits in a cover-object. Redundant bits are those bits that can be modified without corrupting the quality or destroying the integrity of the cover-object.

ii. Embedding process. It selects the subset of the redundant bits to be replaced with data from the information file. The stego-object is created by replacing the selected redundant bits with information file bits. Basically, the model for steganography is shown on Figure 1. The cover-object is a carrier or medium to embed the information file.

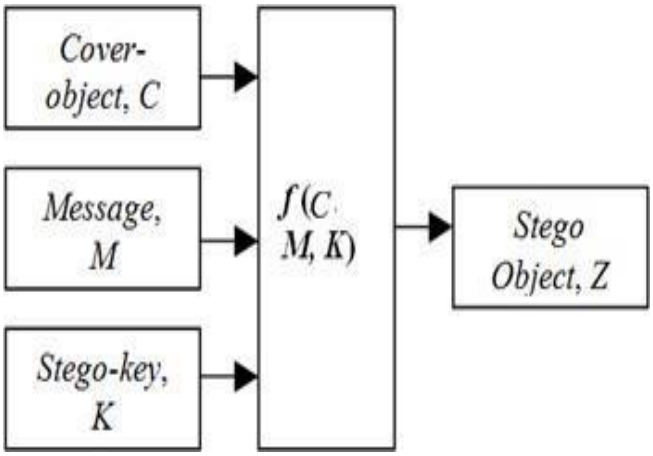

Figure 1. Basic Model of Steganography

2.5 INFORMATION SECURITY:

Information security can be defined as measures adopted to prevent the unauthorized use or modification of use of data or capabilities. For almost all organizations, information is a critical commodity. The various pieces of information used and collected by these organizations, together with the computers that process those pieces of information, are referred to as information assets. **Figure 2** diagrams some of the information assets and where they fit in an organizational hierarchy.

Information can be defined as any Data that is:

i. Accurate and timely.
ii. Specific and organized for a purpose.
iii. Presented within a context that gives it meaning and relevance, and
iv. Can lead to an increase in understanding and decrease in uncertainty.

2.5.1 INFORMATION VALUE

Information is valuable because it can affect behavior, a decision, or an outcome. A piece of information is considered valueless if, after receiving it, things remain unchanged.

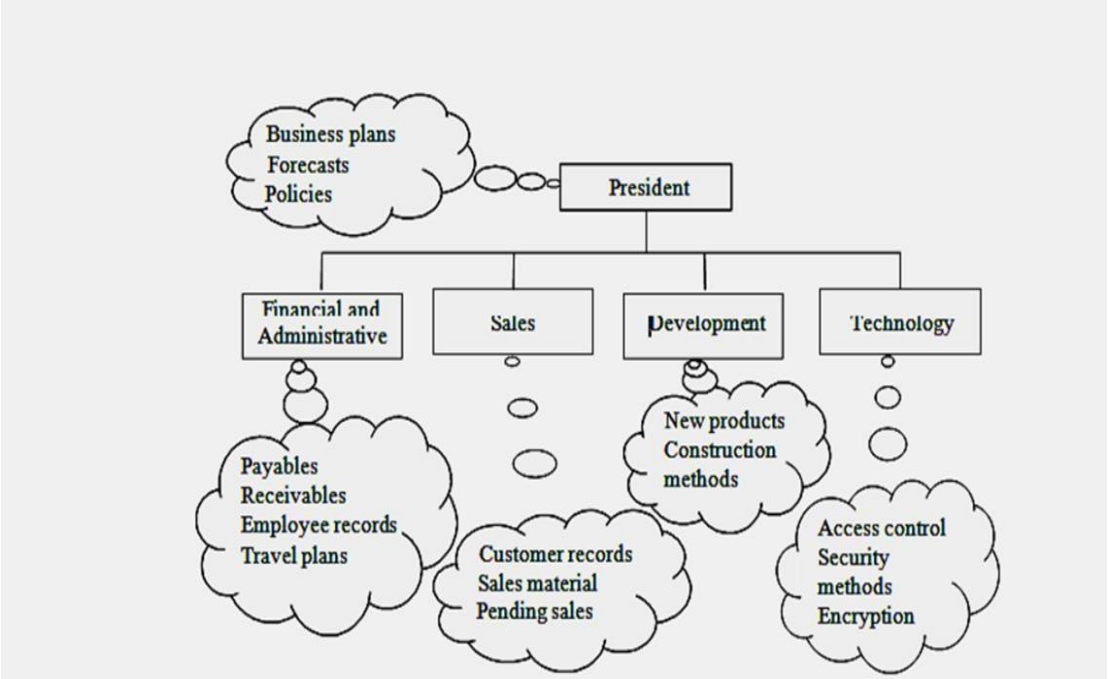

Figure 2. Organization chart with sample information assets

In some cases, organizations spend significant investments gathering or generating information, and then ultimately give the information away for free (for marketing, customer support, reputation building, or participation in professional societies, among other reasons). Instead of the initial investment, the value comes from the use the organization makes of the information.

Other types of organizations do not measure value in terms of money or effort, but in terms of reputation (particularly in government) or operational advantages (particularly in the military, or in nonprofit organizations, although the advantages sought by these organizations differ greatly). Organizational use is partially determined by the properties of the information, and the protections associated with those properties. **Figure 3** shows these properties contributing to information security.

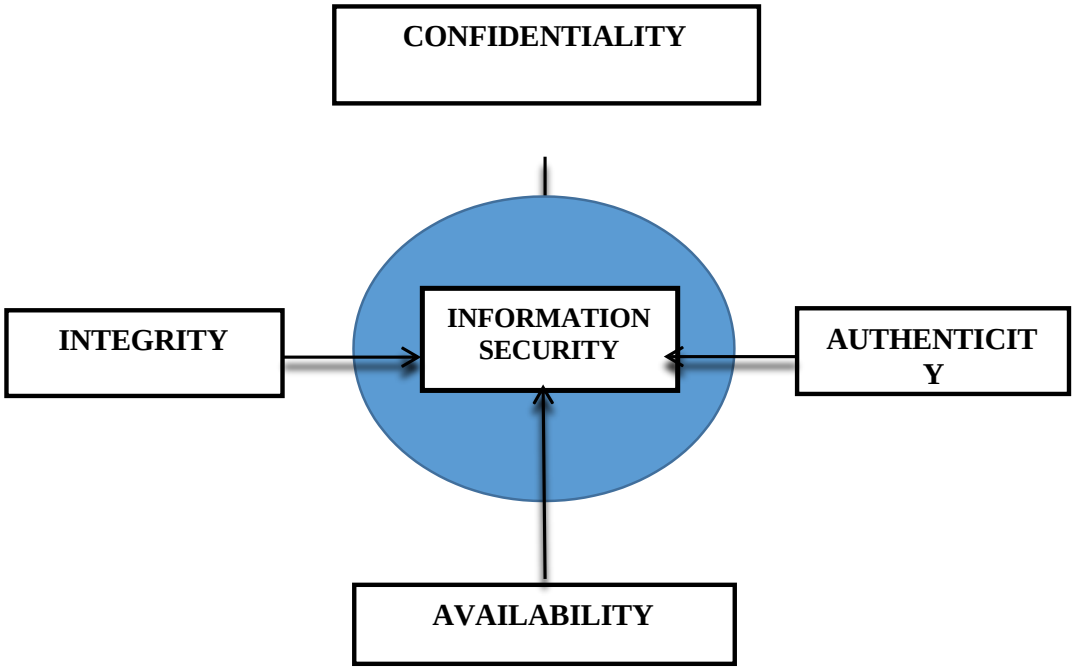

Figure 3. Properties contributing to information security.

2.6 SECURITY CLASSIFICATION

In general, Security is the degree of resistance to, or protection from harm. It applies to any vulnerable and valuable asset, such as a person, dwelling, community, nation, or organization. Security is classified into different layers depending on the type of content intended to be secured, these classification layers are.

i. **Physical security:** Defines the required issues that are needed to protect the physical data or objects from unauthorized intrusion.

ii. **Personal security:** It is defined as the security of the individuals who are officially authorized to access information about the company and its operations.

iii. **Operational security:** It mainly relies on the protection of the information of a particular operation of the chain of activities.

iv. **Communication's security:** The communication's security encompasses the security issues regarding the organization's communication media, technology and content.

v. **Network security:** The network security is responsible for safeguarding the information regarding the networking components, connections and contents.

vi. **Information security:** Information security is the protection of information and the systems and hardware that use, store, and transmit that information.

2.6.1 INFORMATION SECURITY ATTACKS

Information transmitted from a source to an intended and specific destination is known as normal information flow. But an unauthorized user might hack the network or system in order to access or modify the original information. These types of attacks are formally known as security attacks. Unauthorized person can disrupt this normal flow by implementing the different types of techniques over the information and network in following ways.

2.6.1.1 **Interruption:** Interruption is an attack by which the hackers can interrupt the data before reaching the destination. This type of attack shows the effect on availability and usually destroys the system asset and makes the data unavailable or useless.

2.6.1.2 **Interception:** Interception is one of the well-known attacks. When the network is shared that is through a local area network is connected to Wireless LAN or Ethernet it can receive a copy of packets intended for other device. On the internet, the determined hacker can gain access to email traffic and other data transfers. This type of attack shows the effect on confidentiality of data.

2.6.1.3 **Modification:** This refers to altering or replacing of valid data that is needed to send todestination. This type of attacks is done usually by unauthorized access through tampering the data. It shows effect on the integrity of the data.

2.6.1.4 **Fabrication:** In this type, the unauthorized user places data without the interface of source code. The hacker or unauthorized person inserts the unauthorized objects by adding records to the file, insertion of spam messages etc. This type of attack effects on the Authenticity of message.

2.6.2 INFORMATION SECURITY MEASURES

There are many types of security attacks that will try to modify the original information or expose it to unauthorized persons. The main goal of any organization / individual transmitting the information is to implement security measures which include:

2.6.2.1 **Prevention:** The security attacks can be prevented by using an encryption algorithm torestrict any unauthorized access to the encryption algorithm. Then the attacks on confidentiality of the transmitted data will be prevented.

2.6.2.2 **Detection:** Using the intrusion detection systems for detection of unauthorized individuals logged onto a system and making the resources available to legitimate users.

2.6.2.3 **Response:** Whenever the unauthorized attacks happen in the system, the security mechanisms can detect the process and the system can respond to make the data unavailable.

2.6.2.4 **Recovery:** Recovery is the final approach if an attacker modifies the data or makes thedata unavailable. The data can then be recovered by using backup systems, so that the integrity of the data shall not be compromised.

There are different types of approaches for preventing these security attacks. The most useful approaches is encryption. But not just encryption alone because encryption alone can raise suspicion for attack. This is where steganography plays a serious role. Due to its Security through obscurity.

2.7 REQUIREMENTS FOR HIDING INFORMATION DIGITALLY.

There are many different protocols and embedding techniques that enable us to hide information in a given object. However, all of the protocols and techniques must satisfy a number of requirements so that steganography can be applied correctly.

The following is a list of main requirements that steganography techniques must satisfy:

2.7.1 Information integrity.

2.7.2 Information confidentiality.

2.7.3 Information Authenticity.

2.7.4 Information availability.

i. Information Integrity:

The information must have a reliable meaning. This meaning may be protected by assuring the information's data integrity. Preventing undesirable and unauthorized changes to the information . Integrity is the property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. This implies both that control is exercised over the content of information in the system and over modifications made to that information.

By attacking data integrity (i.e., modifying information without authorization), an adversary may block necessary processing, impede other defenses, or disrupt computer-based capabilities.

By protecting integrity, the organization assures that the content, format, and semantics of the information remain controlled by the organization, facilitating ongoing reliable use. For businesses, assuring data integrity protects their ability to receive orders, produce quality products, deliver orders to appropriate addresses, invoice the correct parties for payment, and other aspects of business life. For military organizations, assuring data integrity protects their logistics chain, deployment, targeting, damage assessment, and military operations in general.

ii. Information Confidentiality:

The information must provide an advantage to its use, providing a benefit not available to all. This advantage may derive from the information's confidentiality. Restricting the knowledge of the information to authorized parties.

By attacking confidentiality (i.e., intercepting information without authorization), an adversary may compromise authenticating information, publicize internal planning and forecast information, and disclose personal identifying information (allowing impersonation to other organizations) and other closely held information.

By protecting confidentiality, the organization retains control over which parties have access or use of the information, restricting competitors from exploiting this information, and enabling necessary internal planning. If an organization cannot prevent unauthorized disclosure of information, then it is difficult for that organization to retain control over the use of that information. When the information is critical enough, a clear and unambiguous structure for the analysis of confidentiality becomes useful.

iii. Information Authenticity:

The information must have suitable authority for its use. This authority is assured by the authenticity of the information protected by nonrepudiation, which is the ability to definitively identify the source of the information [22].

By attacking authenticity (i.e., imitating authorized sources), an adversary may cause false information to be accepted as valid, or fraudulently obtain increased access to computers or data.

By protecting authenticity, the organization assures a trace from the responsible individuals to any actions done on the data. This trace enables organizations to audit actions by

individuals, reducing the chance that malice or error will corrupt key elements of the organization's information. For businesses, such authority establishes managerial direction over company processes. For the military, such authority enables the chain of command and reduces uncertainty in decision making.

iv. Information Availability:

The information must be available when needed by the organization. The availability protections ensure that the information and its processing capabilities are providing service as designed whenever users make authorized requests.

Protecting availability involves ensuring the presence of the information, its access rights, usable formatting, and suitable computing resources for handling the information. By attacking availability (i.e., denying service or access), an adversary may block the retrieval, processing, or use of key information supporting an organization's mission. By defending availability, an organization assures that its necessary activities may continue. There are other properties that may be significant to an organization's information security. (Parker D. 2010), for example, adds utility and possession. However, the four listed above give a sense of the variety of ways that information may be attacked and defended. While all of these characteristics of information are required for use in organizations, the importance of each characteristic varies from organization to organization.

In financial institutions, data integrity is paramount, if an institution loses the reliability of its information, regulators will shut it down. In e-commerce (e-businesses), availability is key. Loss of service may lead to large loss of revenue. For many military applications, confidentiality is the most important property, disclosure to the enemy of military plans or operations could be fatal to the personnel's involved. In each of these cases, the value of the information (and thus, the corporate value) is increased via the protection of the information and decreased by lack of such protection

2.8 STEGANOGRAPHY APPLICATIONS:

There are many applications for digital steganography of image, including copyright protection, feature tagging, and secret communication and confidential data storage. Copyright notice or watermark can be embedded inside an image to identify it as intellectual property. Were as confidential information can be hidden in an image to prevent unauthorized view, modification, and use.

2.8.1 DATA STEGANOGRAPHY

The advent of computers has allowed us to begin embedding messages into pictures or sound files. To the human eye, the picture itself remains unchanged, yet within it there could be up to a book's worth of information. Computers, as we may know, operate in binary. That means that every letter and instruction is eventually broken down into a code of '1's and '0's.

Let's say that the binary for the letter 'A' is 11101101. Originally, computer architects designed this system in such a way that the very last '1' or '0' had no particular influence on the value of the designated character. If the last number in this message were '0' instead of '1', the computer would still know that this is an 'A'. 11101100.

The last digit of all binary messages, which is neither meaningful nor necessary, is known as the Least Significant Bit (LSB). One method, used by data steganography software, is to break up the hidden message between the LSBs of the carrier in a pre-determined pattern. This does not change the original meaning of the message. This method implies that the hidden message cannot be bigger than the carrier and should really be much smaller.

2.9 STEGANOGRAPHIC TECHNIQUES:

Over the past few years, numerous steganography techniques that embed hidden messages in multimedia objects have been proposed. There have been many techniques for hiding information or messages in images in such a manner that the alterations made to the image are perceptually indiscernible. Common approaches are including:

2.9.1 Least significant bit insertion (LSB)
2.9.2 Masking and filtering
2.9.3 Transform techniques

i. Least Significant Bits Insertion (LSB)

This is the simple approach to embedding information in image file. This steganographic technique embeds the bits of the message directly into least significant bit plane of the cover-image in a deterministic sequence. Manipulating the least significant bit does not result in human- perceptible difference because the amplitude of the change is small. LSB is always the last bit on the right-hand side of any binary number. Changing this bit causes the least possible effect to the original value.

In a 24-bit image, there are 3 bytes of data to represent RGB values for every pixel in that image. This implies that we can store/hide 3 bits in every pixel. For example, if the image has the following bits:

10010101 00001101 11001001
10010110 00001111 11001010
10011111 00010000 11001011

To store **101101101**. We replace with the original LSBs like this: 1001010**1**
0000110**0** 1100100**1**
1001011**1** 0000111**0** 1100101**1**
1001111**1** 0001000**0** 1100101**1**

To reveal the stored message, the LSBs are extracted alone from the Stego object or image and combined together.

ii. Masking and Filtering Technique

This technique is usually restricted to 24 bits and gray scale images. Hiding information by marking an image, in a manner similar to paper watermarks. The techniques performs analysis of the image, thus embed the information in significant areas so that the hidden message is more integral to the cover image than just hiding it in the noise level.

iii. Transform Techniques

This technique embeds the message by modulating coefficients in a transform domain, such as the Discrete Cosine Transform (DCT) used in JPEG compression, Discrete Fourier Transform, or Wavelet Transform. These methods hide messages in significant areas of the cover-image, which make them more robust to attack. Transformations can be applied over the entire image, to block throughout the image, or other variants.

2.10 IMAGE DEFINITION AND STRUCTURE

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image. This numeric representation forms a grid, and the individual points are referred to as pixels. Most images consist of a rectangular map of the

image's pixels (represented as bits) where each pixel is located and its color. These pixels are displayed horizontally row by row. The number of bits in a color scheme, called the bit depth, refers to the number of bits used for each pixel. The smallest bit depth in current color schemes is 8, meaning that there are 8 bits used to describe the color of each pixel.

2.10.1 IMAGE PROCESSING

A digital image is produced using a camera, scanner or other device. The size of an image can be given in pixels, for example an image which is 640 x 480 pixels contains 307,200 pixels. Each pixel is generally stored as 24-bit or 8-bit. A 24-bit pixel has a possibility of 224 color combinations.

The 24 bits of a 24-bit image are spread over three bytes and each byte represents red, green and blue respectively. Colors are obtained by mixing red, green and blue light in different proportions. An image can be formed by making three measurements of brightness at each pixel using the red, green and blue components of the detected light.

Each pixel gets its own color by combining percentages of red, green and blue (RGB). Each of these colors has a value from 0 to 255.Zero (0) designate that the color is present while 255 designates complete saturation of that color. RGB color has a total of 16,777,216 possible colors. Thus, the total of 255x255x255. Each byte can have a value from 0 to 255 representing the intensity of the color. The darkest color value is 0 and the brightest is 255.

For example, a pixel could be made up of three bytes as follows: 11111111 00000000 00000000. The first 8 bits represent red, the second 8 bits represent green and the third 8 bits represent blue. The bit values in this example result in a red pixel. Its red byte is at a maximum value (11111111) and its green (00000000) and blue (00000000) bytes have the lowest possible value.

Transparency is controlled by the addition of information to each element of the pixel data. This is to allow image overlay. A 24-bit pixel value can be stored in 32 bits. The extra 8 bits specify transparency. This is sometimes called the alpha channel. An ideal 8-bit alpha channel can support transparency levels from 0 (completely transparent) to 255 (completely opaque).

2.10.2 IMAGE FORMATS

The most prominent image formats, exclusively on the internet, are the graphics interchange format (GIF), joint photographic expert's group (JPEG) format, and to a lesser degree, the portable network graphics (PNG) format. The important issue to touch here is that most of the steganographic techniques attempt to exploit the structure of these formats. However, most steganography uses the bitmap format (BMP) simply because of its simple and uncomplicated data structure.

2.10.3 BITMAP IMAGES

Any computer graphics can be categorized either as raster (bitmap image) or as vector image. Bitmap images are most widely used images on the web with exceptions of scalable vector graphics and flash. The group of bitmaps includes icon files, photographs, icons, and any other files that employs pixel to represent an image file. Bitmap images are composed of pixels representing a specific location. Each pixel contains color information.

Any bitmap image is composed of numerous pixels. This is analogous to a body composed of millions of cells. Naturally, a bigger bitmap image has more pixels, and smaller images have lesser number of pixels. Most BMP files have a relatively large file size since the computer has to store information about every single pixel in the image; the file size of a bitmap image is often quite large which serves as a good medium for data hiding, because the bigger the size the more data it can hold.

2.11 IMAGE COMPRESSION

When working with larger images of greater bit depth, the images tend to become too large to transmit over a standard Internet connection. In order to display an image in a reasonable amount of time, techniques must be incorporated to reduce the image's file size. These techniques make use of mathematical formulas to analyze and condense image data, resulting in smaller file sizes. This process is called compression.

In images there are two types of compression: Lossy and lossless. Both methods save storage space, but the procedures that they implement differ. Lossy compression creates smaller files by discarding excess image data from the original image. It removes details that are too small for the human eye to differentiate, resulting in close approximations of the original image, although not an exact duplicate. An example of an image format that uses this compression technique is JPEG (Joint Photographic Experts Group).

Lossless compression, on the other hand, never removes any information from the original image, but instead represents data in mathematical formulas. The original image's integrity is maintained, and the decompressed image output is bit-by-bit identical to the original image input [31]. The most popular image formats that use lossless compression is GIF (Graphical Interchange Format) and 8-bit BMP (Microsoft Windows bitmap file).

Compression plays a very important role in choosing which steganographic algorithm to use. Lossy compression techniques result in smaller image file sizes, but it increases the possibility that the embedded message may be partly lost due to the fact that excess image data will be removed. Lossless compression though, keeps the original digital image intact without the chance of lost, although it does not compress the image to such a small file size.

2.12 DETECTING STEGANOGRAPHY

The art of detecting Steganography is referred to as Steganalysis. To put it simply Steganalysis involves detecting the use of Steganography. Steganalysis does not deal with trying to decrypt the hidden information, just discovering it. There for Steganalysis is "the process of detecting steganography by looking at variances between bit patterns and unusually large file sizes".

One method that can be used to detect Steganography is: Viewing the file and comparing it to another copy of the same file found on the Internet (Picture File.)

CHAPTER-3

RESEARCH METHODOLOGY AND SYSTEM DESIGN

3.0 INTRODUCTION

This chapter provides a detailed explanation on the procedures used in carrying out this study, for the system to work successfully. The chapter also gives an insight into scope and features of the system as well as the requirements. The development tools used and data flow diagrams as well as the system design. The system was designed and will be executed using software engineering methods with strict adherence to the principles of software development life cycle.

3.1 SCOPE OF THE SYSTEM

The system will offer a secure way of encryption of confidential data through obscurity for ISSYS GH. and its business clients. As a case study, the system is designed to cover only the day-to-day activities of the Information Security unit and the communication units of the organization.

3.2 FEATURES OF THE SYSTEM:

The systems graphical user interface is essential to a user. For this reason, the user interface was well thought – through and made easy to use; as well as using tabs and bottoms where appropriate to enhance the user experience and also an image preview panel which gives a clear image of which image is to be encrypted.

3.3 REQUIREMENTS GATHERING TECHNIQUES

The system was designed primarily to encrypt confidential files on computer systems. As a result, the instrument for data collection for this project was through the primary source with which, a critical examination of the various types of information threats which existed in the organization was collected by various open interviews of the personals in that unit.

In order to get a fair idea of the threats that existed and to take the necessary steps in order to curb it and meet the goal for which the system will be developed. This allowed for the collection of first-hand data. On spot observations were also made without the awareness of the personnel and units involved. This allowed for the accurate capturing of a true reflection of the threats to information security in that unit of the organization as well as helping to cross-validate certain responses from the interviews.

3.3.1 FUNCTIONAL REQUIREMENTS

These are the functions that the system must deliver in order to meet user requirements. The functional requirements of this system include:

- The system will allow an effective and reliable way of security for information through obscurity.
- The system will not allow the User(s) to modify encrypted image in any way.
- The system will allow for the use of any image file and information file to be used respectively for encryption.
- User(s) should have a basically fair knowledge in computer file types and computing in order to effectively make use of the system.

3.3.2 NON-FUNCTIONAL REQUIREMENTS

These requirements form the constrain of the system and can be divided into two parts. Hardware and software.

3.3.3 HARDWARE REQUIREMENTS

Since Microsoft .Net framework prepares a huge amount of tool and options for programming. One of .Net tools for pictures and imagery is auto-converting feature. Since the output of the encryption process is a bitmap image (*.bmp). Also the encryption algorithm needs a fast processing time. These technologies need an immerse hardware resources which includes;

- Minimum memories of 1GB RAM.
- A minimum processor speed of 1GHz.
- A minimum video graphic acceleration of 256MB RAM.
- A minimum free hard disk space of 200MB.

3.3.4 SOFTWARE REQUIREMENTS

The system would require:

- 32bit Windows vista, windows 7 and windows 8 Operating system.
- Microsoft .Net framework 4.5.

3.4 PROPOSED SYSTEM ARCHITECTURE

The data hiding patterns using the steganographic technique in this project can be

explained using this simple block diagram. The graphical block representation of this system is as shown in

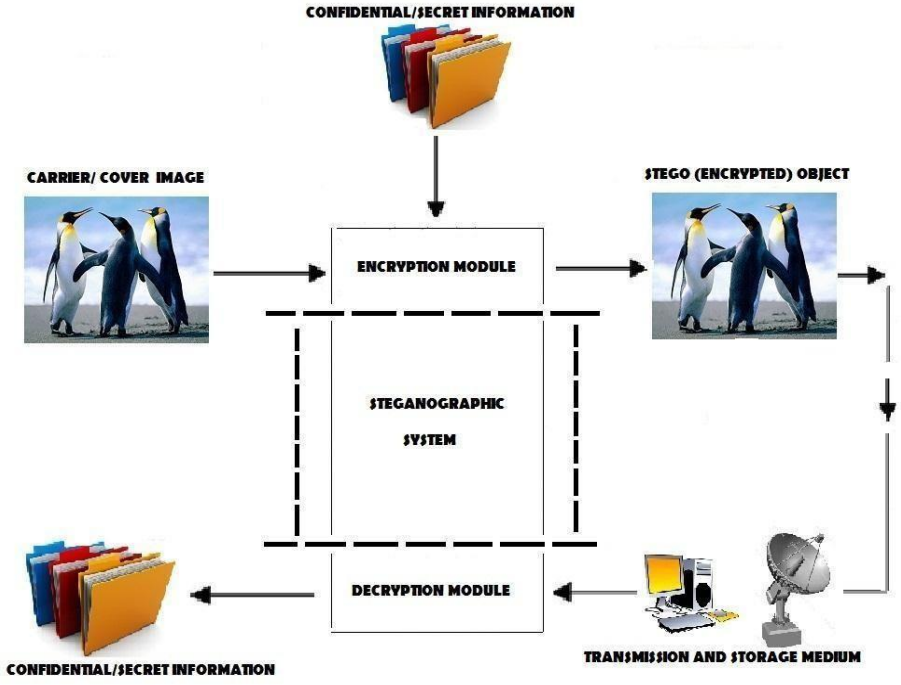

Figure 4. Architecture of proposed model

3.5 SYSTEM DESIGN MODULE PHASES

The systems module designs at different phases are:

- Encryption module phase
- Decryption module phase

(i) Encryption Module Phase

The encryption module phase of the system is the primary stage. In this phase, the user selects the image file, which acts as a carrier or cover image as well as an information file that will be hidden in the carrier image.

In this project, I use Bitmap (*.bmp) images as a default carrier because bitmap images have a large image size and also bitmap images are pixel dependent compared to jpeg image and other image formats. But the user has an option to choose other image file types as carrier image file or cover image file.

In the encryption module, the information file will be embedded into the cover or carrier

image file. The embedding will be done based on the principle of “Least Significant Bit” (LSB) algorithm.

The LSB algorithm uses the least significant bits of each pixel and replace with the significant bits of the information file, such that the information will be encrypted into the carrier image. This process makes the picture not to lose its resolution. The data embedding into image i.e., encryption is implemented using Microsoft visual C#.

(ii) Decryption Module Phase

In the decryption module of the system, the user selects the stego-image from a location. The user then sends the stego-image to the decryption phase. In the decryption phase, the same “Least Significant Algorithm‟ is implemented but in the reverse way. Here the least significant bits from the stego-image are extracted and combined in an order to structure the original information bits.

After successful arrangement, the information file is decrypted from the carrier file and accessed as an original information file. The information file extraction from the carrier image i.e., decryption is implemented using Microsoft Visual C#.

3.6 DATA FLOW DIAGRAMS

Data flow diagrams are the basic building blocks that define the flow of data in a system to the particular destination and difference in the flow when any alteration happens. It makes whole process like a good document and makes simpler and easy to understand for both programmers and non-programmers by dividing into the sub process.

The data flow diagrams are the simple blocks that reveal the relationship between various components of the system and provide high level overview, boundaries of particular system as well as provide detailed overview of system elements.

The data flow diagrams start from source and ends at the destination level i.e., it decomposes from high level to lower levels. The important things to remember about data flow diagrams are: it indicates the data flow for one way but not for loop structures and it doesn’t

indicate the time factors.

This section reveals about the data flow analysis which states about data that have been used, classification of data flow diagrams based on their functions and the other different levels used in the project. The general notations for constructing a block diagram in this project is shown in figure

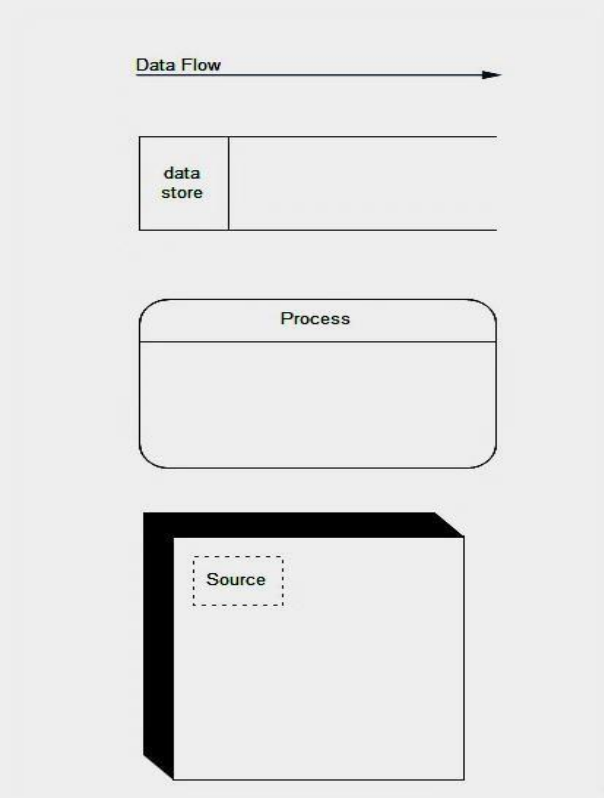

Figure: 6. General representations for constructing a block diagram

i. Data flow processes: It will define the direction i.e., the data flow from one entity to another entity.
ii. Data store: It is the place or physical location where the data is stored after extraction from the data source.
iii. Process: Process defines the source from where the output is generated for the specified input. It states the actions performed on data such that they are transformed, stored or distributed.
iv. Source: It is the starting point or destination point of the data, stating point from where the external entity acts as a cause to flow the data towards destination.

3.6.1 CONSTRUCTING DATA FLOW DIAGRAM

The data flow diagrams can be constructed by dividing the process into different levels like Data Flow Diagram Zero (DFD 0, DFD 1, DFD 2, etc.), for constructing the data flow diagram.

For this process, these simple steps were followed [1];

- The data flow diagram can be constructed only when the process have one dataflow in and one data flow out.
- The process should modify the incoming data and outgoing data.
- The data store should not be alone, should be connected with one process at least.
- The external entities of the process should be involved with one data flow.
- In data process the data flow should be from top to bottom and from left to right.

(i) Data Flow Diagram Level Zero (DFD 0)

This is the highest level view of the system, contains only one process which represents the whole function of the system. It doesn't contain any data stores and the data is stored within the process. For constructing DFD level 0 diagram for the proposed system, two sources are needed; one is for source and another is for destination and a process.

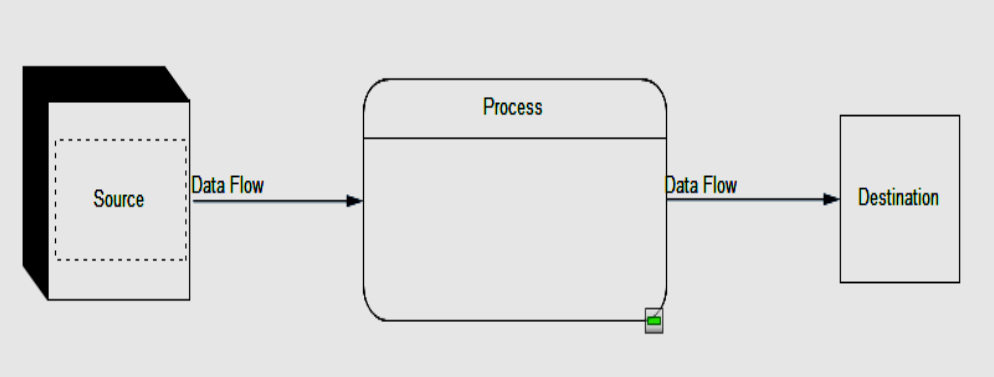

Figure:6. (DFD 0)

Data Flow Zero (DFD 0) is the basic data flow process; the main objective is to transfer the data from user to the save folder destination after encryption.

(ii) Data Flow Diagram Level One (DFD 1)

For constructing (DFD 1) we need to identify and draw the process that makes the level 0 process. In this system, the transferring of the information file from saved location to the new

decrypted location save destination, the information is first encrypted and latter decrypted.

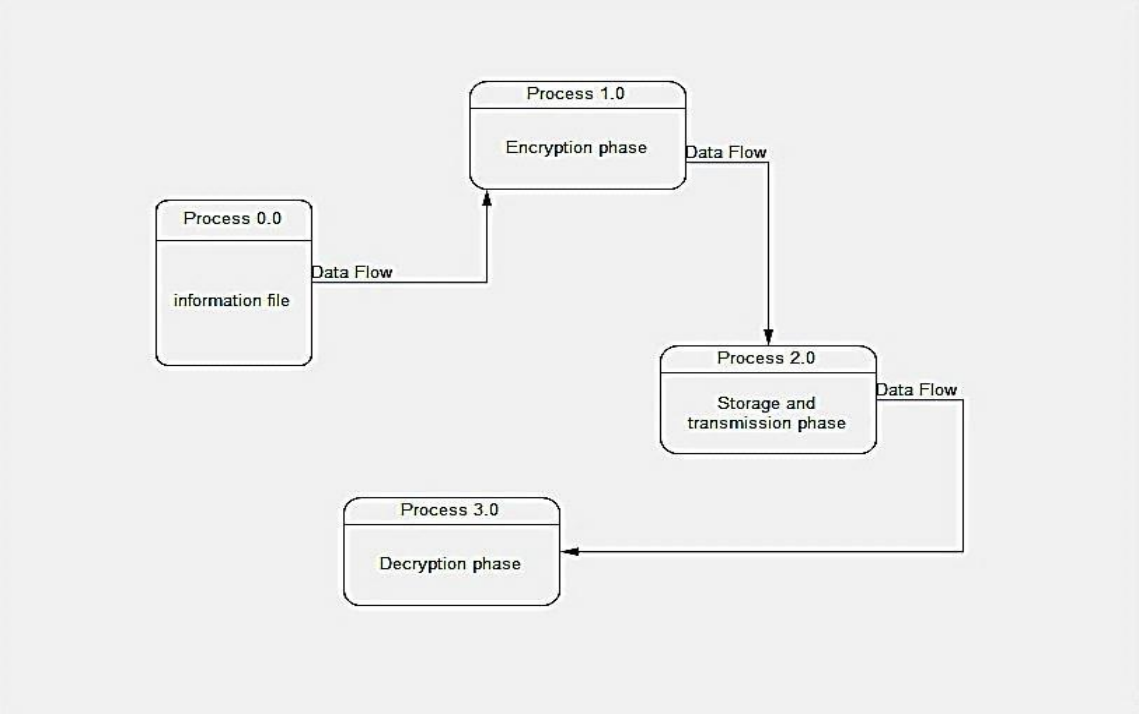

Figure: 7. (DFD 1)

In this data flow diagram, the information file is sent to the encryption phase for embedding it into the cover image for generating the stego-image. In the next phase, the carrier image is sent to the decryption phase through the transmission phase. The final phase is the decryption phase where the information file is extracted from the stego-image and the original information file is saved to a location.

(iii) Data Flow Diagram Level Two (DFD 2)

The image file and the information file are given to the encryption phase. The encryption algorithm is used for embedding the information file into the carrier image file. The resultant stego-image file acting as a carrier image is transmitted to the decryption phase using the transmission medium. For extracting the message from the carrier image, it is sent to the decryption section. The information file is extracted from the stego-image using the decryption algorithm.

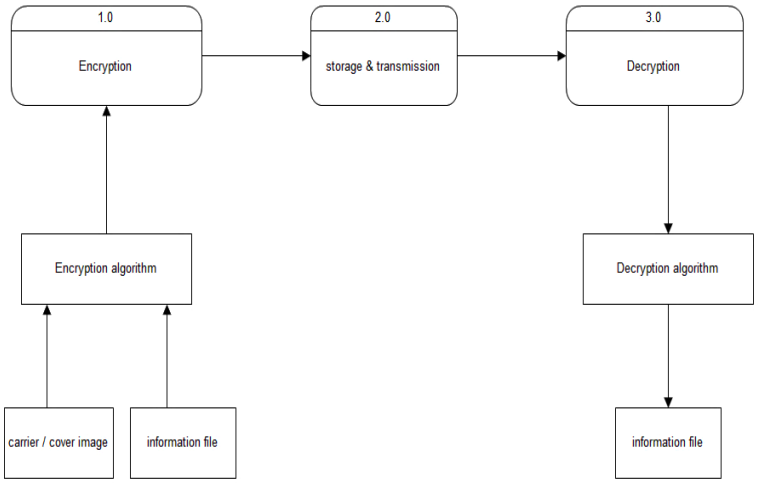

Figure 8. (DFD 2)

CHAPTER 4

IMPLEMENTATION

4.1 SYSTEM IMPLEMENTATION

The system design is translates to programming codes using Microsoft Visual studio C# programming language. Microsoft visual studio C# comes with an inbuilt code analyzer, this automated testing of programming code for the purpose of debugging an application before it is distributed or sold. Code analysis consists of statements created with a text editor or visual programming tool and then saved in a file.

The code is the most permanent form of a program, even though the program may later be modified, improved or upgraded. The code analysis can be either static or dynamic.

i. In static analysis, debugging is done by examining the code without actually executing the program. This can reveal errors at an early stage in program development, often eliminating the need for multiple revisions later.

ii. Dynamic analysis is performed in an effort to uncover more subtle defects or vulnerabilities. Dynamic analysis consists of real-time program testing.

A major advantage of these methods of code analysis is the fact that it does not require developers to make educated guesses at situations likely to produce errors. Other advantages include eliminating unnecessary program components and ensuring that the program under test is compatible with other programs likely to be run concurrently

4.2 CODE

The project was completely implemented in Java Programming. The frontend was designed using JAVA AWT & JAVA SWING, the algorithm was implemented using the standard java programming and both the frontend and backend are integrated together to get the final output.

Main.java

Main.java

steganography > src > Main.java

```java
public class Main {
    Run | Debug
    public static void main(String arg[]) {
        SplashScreenFrame a=new SplashScreenFrame();
        a.setVisible(b: true);
        a.setLocationRelativeTo(c: null);
    }
}
```

Encryption.java

```java
//EmbedMessage.java
 import java.awt.image.*;
 import javax.swing.*;
 import java.awt.*;
 import java.awt.event.*;
 import javax.imageio.*;

 public class Encryption extends JFrame implements ActionListener
 {
 JButton open = new JButton("Open"), embed = new JButton("Embed"),
    save = new JButton("Save into new file"), reset = new JButton("Reset");

 JTextArea message = new JTextArea(10,3);
 BufferedImage sourceImage = null, embeddedImage = null;
 JSplitPane sp = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT);
 JScrollPane originalPane = new JScrollPane(),
    embeddedPane = new JScrollPane();
```

```
 public Encryption() {
 super("Embed stegonographic message in image");
 assembleInterface(); open.setBackground(Color.black);
 open.setForeground(Color.WHITE); open.setFont(new
 Font("Monaco", Font.BOLD, 20));

 embed.setBackground(Color.black);
 embed.setForeground(Color.WHITE); embed.setFont(new
 Font("Monaco", Font.BOLD, 20));

 save.setBackground(Color.black);
 save.setForeground(Color.WHITE); save.setFont(new
 Font("Monaco", Font.BOLD, 20));

 reset.setBackground(Color.black);
 reset.setForeground(Color.WHITE); reset.setFont(new
 Font("Monaco", Font.BOLD, 20));

// this.setBounds(GraphicsEnvironment.getLocalGraphicsEnvironment().
//          getMaximumWindowBounds());
 this.setSize(1000, 700);
 this.setLocationRelativeTo(null);
 this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
 this.setVisible(true); sp.setDividerLocation(0.5);
 this.validate();
 }

 private void assembleInterface() {
 JPanel p = new JPanel(new FlowLayout());
 p.add(open);
 p.add(embed);
 p.add(save);
 p.add(reset);
 this.getContentPane().add(p, BorderLayout.SOUTH);
open.addActionListener(this);
embed.addActionListener(this);
save.addActionListener(this);
reset.addActionListener(this); open.setMnemonic('O');
 embed.setMnemonic('E');
 save.setMnemonic('S');
 reset.setMnemonic('R');

 p = new JPanel(new GridLayout(1,1));
 p.add(new JScrollPane(message));
 message.setFont(new Font("Arial",Font.BOLD,20));
                                                    };
```

```
    p.setBorder(BorderFactory.createTitledBorder("Message to be embedded"));
    this.getContentPane().add(p, BorderLayout.NORTH);

    sp.setLeftComponent(originalPane); sp.setRightComponent(embeddedPane);
    originalPane.setBorder(BorderFactory.createTitledBorder("Original
    Image"));
    embeddedPane.setBorder(BorderFactory.createTitledBorder
    ("Steganographed Image"));
    this.getContentPane().add(sp, BorderLayout.CENTER);
    }

    public void actionPerformed(ActionEvent ae) { Object
    o = ae.getSource();
    if(o == open) openImage();
    else if(o == embed)
       embedMessage();
    else if(o == save)
       saveImage();
    else if(o == reset)
       resetInterface();
    }

    private java.io.File showFileDialog(final boolean open) {
    JFileChooser fc = new JFileChooser("Open an image");
    javax.swing.filechooser.FileFilter ff = new
    javax.swing.filechooser.FileFilter() {
       public boolean accept(java.io.File f) { String name
          = f.getName().toLowerCase(); if(open)
             return f.isDirectory() || name.endsWith(".jpg") || name.endsWith
             (".jpeg") ||
                name.endsWith(".png") || name.endsWith(".gif") ||
       name.endsWith(".tiff") ||
                name.endsWith(".bmp") || name.endsWith(".dib"); return
          f.isDirectory() || name.endsWith(".png")
        ||     name.endsWith(".bmp");
          }
       public String getDescription() { if(open)
             return "Image (*.jpg, *.jpeg, *.png, *.gif, *.tiff, *.bmp,
*.dib)";


        return "Image (*.png, *.bmp)";
    }
    fc.setAcceptAllFileFilterUsed(false);
    fc.addChoosableFileFilter(ff);

    java.io.File f = null;
```

```
  if(open && fc.showOpenDialog(this) == fc.APPROVE_OPTION) f =
      fc.getSelectedFile();
   else if(!open && fc.showSaveDialog(this) == fc.APPROVE_OPTION) f =
      fc.getSelectedFile();
   return f;
   }

private void openImage() {
   java.io.File f = showFileDialog(true); try {
      sourceImage = ImageIO.read(f);
      JLabel l = new JLabel(new ImageIcon(sourceImage));
      originalPane.getViewport().add(l); this.validate();
      } catch(Exception ex) { ex.printStackTrace(); }
   }

private void embedMessage() { String mess
   = message.getText();
   embeddedImage = sourceImage.getSubimage(0,0,
      sourceImage.getWidth(),sourceImage.getHeight());
   embedMessage(embeddedImage, mess);
   JLabel l = new JLabel(new ImageIcon(embeddedImage));
   embeddedPane.getViewport().add(l);
   this.validate();
   }

private void embedMessage(BufferedImage img, String mess) { int
   messageLength = mess.length();

   int imageWidth = img.getWidth(), imageHeight = img.getHeight(),
      imageSize = imageWidth * imageHeight;
   if(messageLength * 8 + 32 > imageSize) {
      JOptionPane.showMessageDialog(this, "Message is too long for the chosen image",
         "Message too long!", JOptionPane.ERROR_MESSAGE); return;
      }
   embedInteger(img, messageLength, 0, 0);

   byte b[] = mess.getBytes(); for(int
   i=0; i<b.length; i++)
      embedByte(img, b[i], i*8+32, 0);
   }

private void embedInteger(BufferedImage img, int n, int start, int
storageBit) {
   int maxX = img.getWidth(), maxY = img.getHeight(),
      startX = start/maxY, startY = start - startX*maxY, count=0; for(int
   i=startX; i<maxX && count<32; i++) {
```

```
      for(int j=startY; j<maxY && count<32; j++) {
         int rgb = img.getRGB(i, j), bit = getBitValue(n, count); rgb =
         setBitValue(rgb, storageBit, bit);
         img.setRGB(i, j, rgb); count++;
         }
      }
   }

private void embedByte(BufferedImage img, byte b, int start, int storageBit)
{
   int maxX = img.getWidth(), maxY = img.getHeight(),
      startX = start/maxY, startY = start - startX*maxY, count=0; for(int
   i=startX; i<maxX && count<8; i++) {
      for(int j=startY; j<maxY && count<8; j++) {
         int rgb = img.getRGB(i, j), bit = getBitValue(b, count); rgb =
         setBitValue(rgb, storageBit, bit);
         img.setRGB(i, j, rgb); count++;
         }
      }
   }

private void saveImage() { if(embeddedImage ==
   null) {
      JOptionPane.showMessageDialog(this, "No message has been embedded!", "Nothing
        to save", JOptionPane.ERROR_MESSAGE);
      return;
     }
  java.io.File f = showFileDialog(false); String
   name = f.getName();
   String ext = name.substring(name.lastIndexOf(".")+1).toLowerCase();
   if(!ext.equals("png") && !ext.equals("bmp") &&   !ext.equals("dib")) {
         ext = "png";
         f = new java.io.File(f.getAbsolutePath()+".png");
         }
   try {
      if(f.exists()) f.delete(); ImageIO.write(embeddedImage,
      ext.toUpperCase(), f);
      } catch(Exception ex) { ex.printStackTrace(); }
   }

private void resetInterface() {
   message.setText("");
   originalPane.getViewport().removeAll();
   embeddedPane.getViewport().removeAll();
   sourceImage = null;
   embeddedImage = null;
   sp.setDividerLocation(0.5); this.validate();
```

```
    }

 private int getBitValue(int n, int location) {
    int v = n & (int) Math.round(Math.pow(2, location));
    return v==0?0:1;
    }

 private int setBitValue(int n, int location, int bit) {
    int toggle = (int) Math.pow(2, location), bv = getBitValue(n, location);
    if(bv == bit)
       return n;
    if(bv == 0 && bit == 1)
       n |= toggle;
    else if(bv == 1 && bit == 0)
       n ^= toggle;
    return n;
    }

// public static void main(String arg[]) {
////     Encryption embedMessage = new Encryption();
//    }
}
```

Decryption.java

```
//DecodeMessage.java
 import java.awt.image.*;
 import javax.swing.*;
 import java.awt.*;
 import java.awt.event.*;
 import javax.imageio.*;

 public class Decryption extends JFrame implements ActionListener
 {
 JButton open = new JButton("Open"), decode = new JButton("Decode"),
    reset = new JButton("Reset");
 JTextArea message = new JTextArea(10,3);
 BufferedImage image = null;
 JScrollPane imagePane = new JScrollPane();

 public Decryption() {
    super("Decode stegonographic message in image");
    assembleInterface();
    this.setSize(800, 600);
    this.setLocationRelativeTo(null);
    this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
 //   this.setBounds(GraphicsEnvironment.getLocalGraphicsEnvironment().
   //    getMaximumWindowBounds());
    this.setVisible(true);
```

```
    open.setBackground(Color.black);
    open.setForeground(Color.WHITE); open.setFont(new
    Font("Monaco", Font.BOLD, 20));

    decode.setBackground(Color.black);
    decode.setForeground(Color.WHITE); decode.setFont(new
    Font("Monaco", Font.BOLD, 20));

    reset.setBackground(Color.black);
    reset.setForeground(Color.WHITE); reset.setFont(new
    Font("Monaco", Font.BOLD, 20));

    }

private void assembleInterface() {
    JPanel p = new JPanel(new FlowLayout());
    p.add(open);
    p.add(decode); p.add(reset);
    this.getContentPane().add(p, BorderLayout.NORTH);
    open.addActionListener(this);
    decode.addActionListener(this);
    reset.addActionListener(this); open.setMnemonic('O');
    decode.setMnemonic('D');
    reset.setMnemonic('R');

    p = new JPanel(new GridLayout(1,1));
    p.add(new JScrollPane(message));
    message.setFont(new Font("Arial",Font.BOLD,20));
    p.setBorder(BorderFactory.createTitledBorder("Decoded message"));
    message.setEditable(false);
    this.getContentPane().add(p, BorderLayout.SOUTH);

    imagePane.setBorder(BorderFactory.createTitledBorder("Steganographed Image"));
   this.getContentPane().add(imagePane, BorderLayout.CENTER);
    }
public void actionPerformed(ActionEvent ae) { Object o
    = ae.getSource();
    if(o == open) openImage();
    else if(o == decode)
       decodeMessage();
    else if(o == reset)
       resetInterface();
    }

private java.io.File showFileDialog(boolean open) {
```

```
 JFileChooser fc = new JFileChooser("Open an image");
 javax.swing.filechooser.FileFilter ff = new
 javax.swing.filechooser.FileFilter() {
    public boolean accept(java.io.File f) { String
       name = f.getName().toLowerCase();
       return f.isDirectory() ||   name.endsWith(".png") ||
    name.endsWith(".bmp");
       }
    public String getDescription() { return
       "Image (*.png, *.bmp)";
       }
    };
 fc.setAcceptAllFileFilterUsed(false);
 fc.addChoosableFileFilter(ff);

 java.io.File f = null;
 if(open && fc.showOpenDialog(this) == fc.APPROVE_OPTION) f =
    fc.getSelectedFile();
 else if(!open && fc.showSaveDialog(this) == fc.APPROVE_OPTION) f =
    fc.getSelectedFile();
 return f;
}

private void openImage() {
 java.io.File f = showFileDialog(true); try {
    image = ImageIO.read(f);
    JLabel l = new JLabel(new ImageIcon(image));
   imagePane.getViewport().add(l); this.validate();
    } catch(Exception ex) { ex.printStackTrace(); }
 }

 private void decodeMessage() { if(image
  == null){
      JOptionPane.showMessageDialog(null, "first open a picture"); return;
  }
 int len = extractInteger(image, 0, 0); byte
 b[] = new byte[len];
 for(int i=0; i<len; i++)
    b[i] = extractByte(image, i*8+32, 0); message.setText(new
 String(b));
 }

private int extractInteger(BufferedImage img, int start, int storageBit) { int
maxX = img.getWidth(), maxY = img.getHeight(),
    startX = start/maxY, startY = start - startX*maxY, count=0; int
 length = 0;
 for(int i=startX; i<maxX && count<32; i++) {
```

```
   for(int j=startY; j<maxY && count<32; j++) {
      int rgb = img.getRGB(i, j), bit = getBitValue(rgb, storageBit); length =
      setBitValue(length, count, bit);
      count++;
      }
   }
return length;
}

private byte extractByte(BufferedImage img, int start, int storageBit) { int
maxX = img.getWidth(), maxY = img.getHeight(),
   startX = start/maxY, startY = start - startX*maxY, count=0; byte b =
0;
for(int i=startX; i<maxX && count<8; i++) { for(int
   j=startY; j<maxY && count<8; j++) {
      int rgb = img.getRGB(i, j), bit = getBitValue(rgb, storageBit); b =
      (byte)setBitValue(b, count, bit);
      count++;
      }
   }
return b;
}

private void resetInterface() {
message.setText("");
imagePane.getViewport().removeAll(); image =
null;
this.validate();
}

private int getBitValue(int n, int location) {
int v = n & (int) Math.round(Math.pow(2, location)); return
v==0?0:1;
}

private int setBitValue(int n, int location, int bit) {
int toggle = (int) Math.pow(2, location), bv = getBitValue(n, location); if(bv ==
bit)
   return n;
if(bv == 0 && bit == 1) n |=
   toggle;
else if(bv == 1 && bit == 0) n ^=
   toggle;
return n;
}

// public static void main(String arg[]) {
   //     Decryption newClass = new Decryption();
  //      }}
```

Menuframe.java

```
import java.awt.Color;

public class MenuFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public MenuFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */ @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel2 = new javax.swing.JPanel(); jButton2 =
        new javax.swing.JButton(); jButton1 = new
        javax.swing.JButton(); jPanel1 = new
        javax.swing.JPanel(); jLabel1 = new
        javax.swing.JLabel(); jPanel3 = new
        javax.swing.JPanel(); jLabel2 = new
        javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("IMAGE STEGNOGRAPHY");

        jPanel2.setBackground(new java.awt.Color(153, 153, 153));




1, 1));
```

```
    jButton2.setBackground(new java.awt.Color(223, 223, 223));
    jButton2.setFont(new java.awt.Font("Algerian", 1, 36)); // NOI18N
    jButton2.setForeground(new java.awt.Color(32, 32, 32));
    jButton2.setText("ENCODE");
    jButton2.setBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1,

    jButton2.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
    jButton2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseEntered(java.awt.event.MouseEvent evt) {
jButton2MouseEntered(evt);

        }
    });
    jButton2.addActionListener(new java.awt.event.ActionListener() { public void
        actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jButton1.setBackground(new java.awt.Color(224, 227, 225));
    jButton1.setFont(new java.awt.Font("Algerian", 1, 36)); // NOI18N
    jButton1.setForeground(new java.awt.Color(23, 23, 23));
    jButton1.setText("DECODE");
    jButton1.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
    jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            jButton1MouseEntered(evt);
        }
    });
    jButton1.addActionListener(new java.awt.event.ActionListener() { public void
        actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jPanel1.setBackground(new java.awt.Color(46, 46, 46));

    jLabel1.setFont(new java.awt.Font("Sitka Banner", 1, 36)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(254, 254, 254));
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("IMAGE STEGNOGRAPHY ");
```

```
  javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
  jPanel1.setLayout(jPanel1Layout);
  jPanel1Layout.setHorizontalGroup(

  jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
  jPanel1Layout.createSequentialGroup()
          .addContainerGap()
  .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, 653,
Short.MAX_VALUE)
          .addContainerGap())
  );
  jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen.addGroup(jPane
l1Layout.createSequentialGroup()
  .addGap(26, 26, 26  t.LEADING)
  .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addContainerGap(22, Short.MAX_VALUE))
  );

  jPanel3.setBackground(new java.awt.Color(0, 102, 102));

  jLabel2.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
  jLabel2.setForeground(new java.awt.Color(255, 255, 255));
  jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
  jLabel2.setText("DESIGNED AND DEVELOPED BY SWAROOP");

  javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
  jPanel3.setLayout(jPanel3Layout);
  jPanel3Layout.setHorizontalGroup(
  jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
  t.LEADING)
```

```
    .addGroup(jPanel3Layout.createSequentialGroup()
    .addContainerGap()
    .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
    Javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
    );
    jPanel3Layout.setVerticalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    jPanel3Layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)
    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
    javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(27, 27, 27))
    );

 javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
    t.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
    .addGap(175, 175, 175)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupL
    ayout.Alignment.LEADING, false)
                .addComponent(jButton2,
    javax.swing.GroupLayout.DEFAULT_SIZE, 326, Short.MAX_VALUE)
    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap()
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
  Alignment.LEADING)
 .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
 javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
 .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
 javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap())
    );
    jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
    t.LEADING)
```

```
.addGroup(jPanel2Layout.createSequentialGroup()
 .addContainerGap()
.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addGap(43, 43, 43)
              .addComponent(jButton2,
     javax.swing.GroupLayout.PREFERRED_SIZE, 51,
     javax.swing.GroupLayout.PREFERRED_SIZE)
              .addGap(43, 43, 43)
              .addComponent(jButton1,
     javax.swing.GroupLayout.PREFERRED_SIZE, 52,
     javax.swing.GroupLayout.PREFERRED_SIZE)
     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RE LATED, 45,
     Short.MAX_VALUE)
              .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
     javax.swing.GroupLayout.PREFERRED_SIZE)
              .addGap(23, 23, 23))
     );

     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
     getContentPane().setLayout(layout);
     layout.setHorizontalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
             .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
     layout.createSequentialGroup()
              .addContainerGap()
              .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
     javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
              .addContainerGap())
```

```
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
    );

    setSize(new java.awt.Dimension(729, 484));
    setLocationRelativeTo(null);
}// </editor-fold>//GEN-END:initComponents

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton2ActionPerformed
    Encryption a=new Encryption();

            a.setVisible(true);
}//GEN-LAST:event_jButton2ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
    Decryption a=new Decryption();

            a.setVisible(true);
}//GEN-LAST:event_jButton1ActionPerformed

private void jButton2MouseEntered(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_jButton2MouseEntered

    jButton2.setBackground(Color.lightGray);
}//GEN-LAST:event_jButton2MouseEntered

private void jButton1MouseEntered(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_jButton1MouseEntered

    jButton1.setBackground(Color.LIGHT_GRAY);
}//GEN-LAST:event_jButton1MouseEntered

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {/* Set the Nimbus look and feel
*///<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
    (optional) ">
```

```
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
        default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */ try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(MenuFrame.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(MenuFrame.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(MenuFrame.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(MenuFrame.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MenuFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify//GEN-BEGIN:variables private
    javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2; private
    javax.swing.JLabel jLabel1; private
    javax.swing.JLabel jLabel2; private
    javax.swing.JPanel jPanel1; private
    javax.swing.JPanel jPanel2; private
    javax.swing.JPanel jPanel3;
    // End of variables declaration//GEN-END:variables
}
```

CHAPTER – 5

TESTING

TESTING

Testing is the process of evaluation a software item to detect differences between given input and expected output. It is also to assess the features of the software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process after implementation (coding). In other words software testing is a verification and validation process.

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

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

5.1 BASICS OF SOFTWARE TESTING

In order to start the testing process the primary thing is requirements of software development cycle. Using this phase the testing phase will be easier for testing.

The capacity of the software can be calculated by executing the code and inspecting the code in different conditions such as testing the software by subjecting it to different sources as input and examining the results with respect to the inputs.

There are two basics of software testing: black box testing and white box testing:

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

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

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

5.2 FUNCTIONAL AND NON-FUNCTIONAL TESTING

5.2.1 Functional testing: Defines the specified function of a particular code in the program.This type of testing gives us a brief description about the program's performance and security in the various functional areas.

5.2.2 Non-functional testing: Defines the capabilities of particular software like its log dataetc. It is opposite to functional testing and so will not describe the specifications like security and performance.

The performance of the particular program not only depends on errors in coding. The errors in the code can be noticed during execution, but the other types of errors can affect the program performance like when the program is developed based on one platform that may not perform well and give errors when executed in different platform. So, compatibility is another issue that reduces the software performance.

5.3 AIM OF TESTING

The main aim of testing is to analyze the performance and to evaluate the errors that occur when the program is executed with different input sources and running indifferent operating environments. There are different types of approaches for testing a .NET framework based application are;

- Unit testing
- Validation testing
- Integration testing
- User acceptance testing
- Output testing
- Black box and white box testing.

i. Unit Testing:

This is the approach of taking a small part of testable application and executing it according to the requirements and testing the application behavior. Unit testing is used for detecting the defects that occur during execution (MSDN, 2010).When an algorithm is executed, the integrity should be maintained by the data structures. Unit testing is made use for testing the functionality of each algorithm during execution. Unit testing reduces the ambiguity

in the units.

In this project, I have developed an application using different phases like encryption, decryption. So for getting the correct output, all the functions that are used are executed and tested at least once so as to making sure that all the control paths, error handling and control structures are in proper manner.

ii. **Limitations of Unit Testing**: This is limited to test only the functionality of the units. It can't identify integration errors, performance problems and system problems. Unit testing can show the errors which occur in the units when the testing runs. It may not display the errors that currently are absent.

iii. **Validation Testing:** Validation is the process of finding whether the product is built correct or not. The software application or product that is designed should fulfill the requirements and reach the expectations set by the user. Validation is done while developing or at the final stage of development process to determine whether it is satisfies the specified requirements of user.

Using validation test the developer can qualify the design, performance and its operations. Also the accuracy, repeatability, selectivity, Limit of detection and quantification can be specified using "Validation testing" (MSDN, 2010).

i. **Output Testing:** After completion of validation testing the next process is output testing. Output testing is the process of testing the output generated by the application for the specified inputs. This process checks weather the application is producing the required output as per the user's specification or not. The "output testing" can be done by considering mainly by updating the test plans, the behavior of application with different type of inputs and with produced outputs, making the best use of the operating capacity and considering the recommendations for fixing the issues (MSDN, 2010).

ii. **Integration Testing:** This is an extension to unit testing, after unit testing the units are integrated with the logical program. The integration testing is the process of examining the working behavior of the particular unit after embedding with program. This procedure

identifies the problems that occur during the combination of units. The integration testing can be normally done in three approaches;

- Top-down approach
- Bottom-up approach
- Umbrella approach

(a) Top-down approach:

In the top-down approach the highest-level module should be considered first and integrated. This approach makes the high-level logic and data flow to test first and reduce the necessity of drivers. One disadvantage with top-down approach is its poor support and functionality is limited (MSDN, 2010)

(b) Bottom-up approach:

Bottom-up approach is opposite to top-down approach. In this approach, the lowest level units are considered and integrated first. Those units are known as utility units. The utility units are tested first so that the usage of stubs is reduced. The disadvantage in this method is that it needs the respective drivers which make the test complicated, the support is poor and the functionality is limited (MSDN, 2010).

(c) Umbrella approach:

The third approach is umbrella approach, which makes use of both the top – bottom and bottom - top approaches. This method tests the integration of units along with its functional data and control paths. After using the top - bottom and bottom-top approaches, the outputs are integrated in top - bottom manner.

The advantage of this approach is that it provides good support for the release of limited functionality as well as minimizing the needs of drivers and hubs. The main disadvantage is that it is less systematic than the other two approaches (MSDN, 2010).

(iii) User Acceptance Testing

This is the process of obtaining the confirmation from the user that the system meets the set of specified requirements. It is the final stage of project; the user performs various tests during the design of the applications and makes further modifications according to the requirements to achieve the final result. The user acceptance testing gives the confidence to the clients about the performance of system.

5.4 TEST CASES

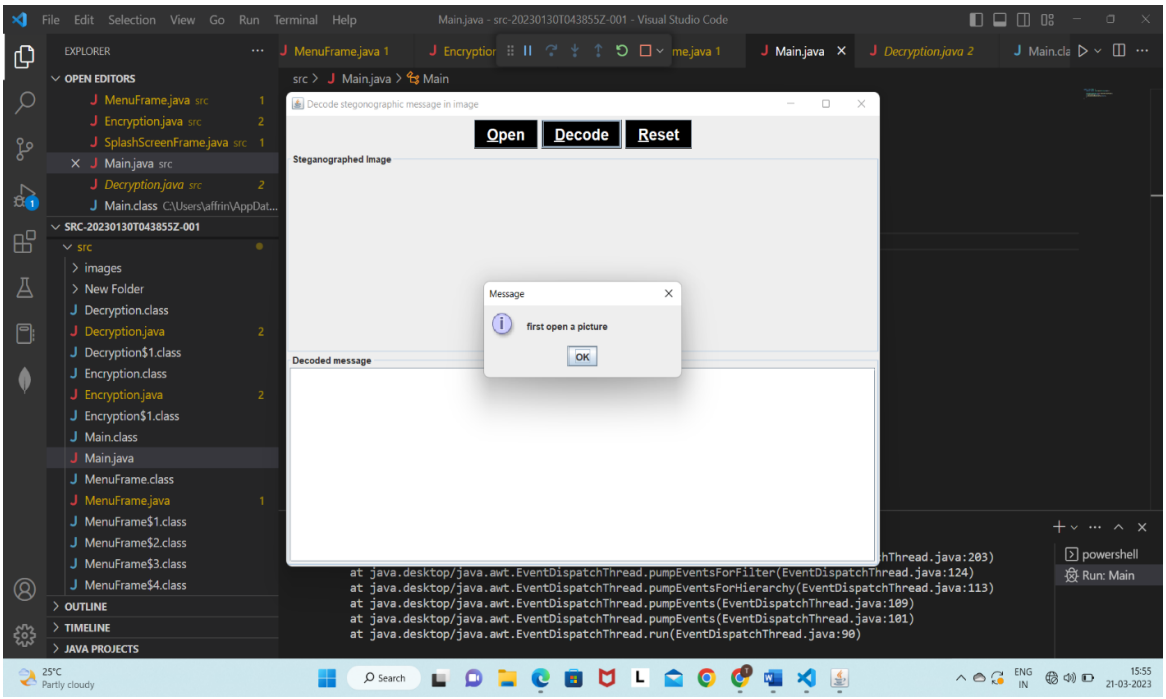

Figure 9.Decoding without Encoding

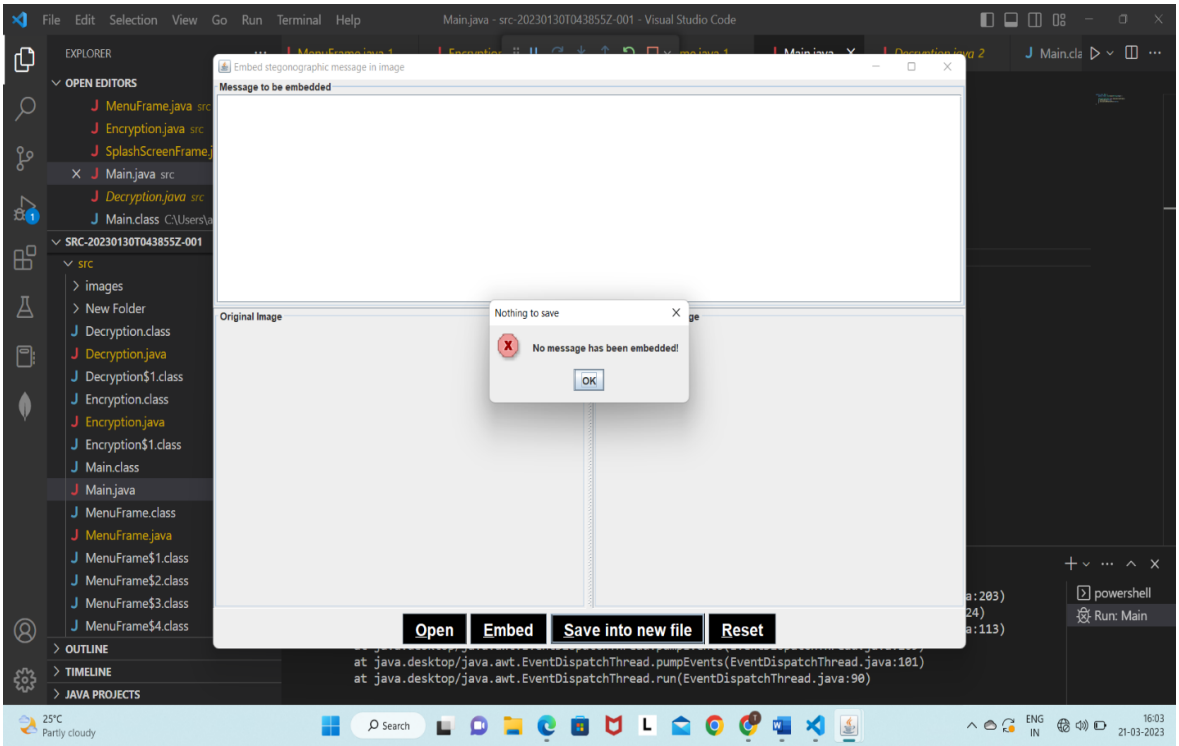

Figure 10.Embedding without a message

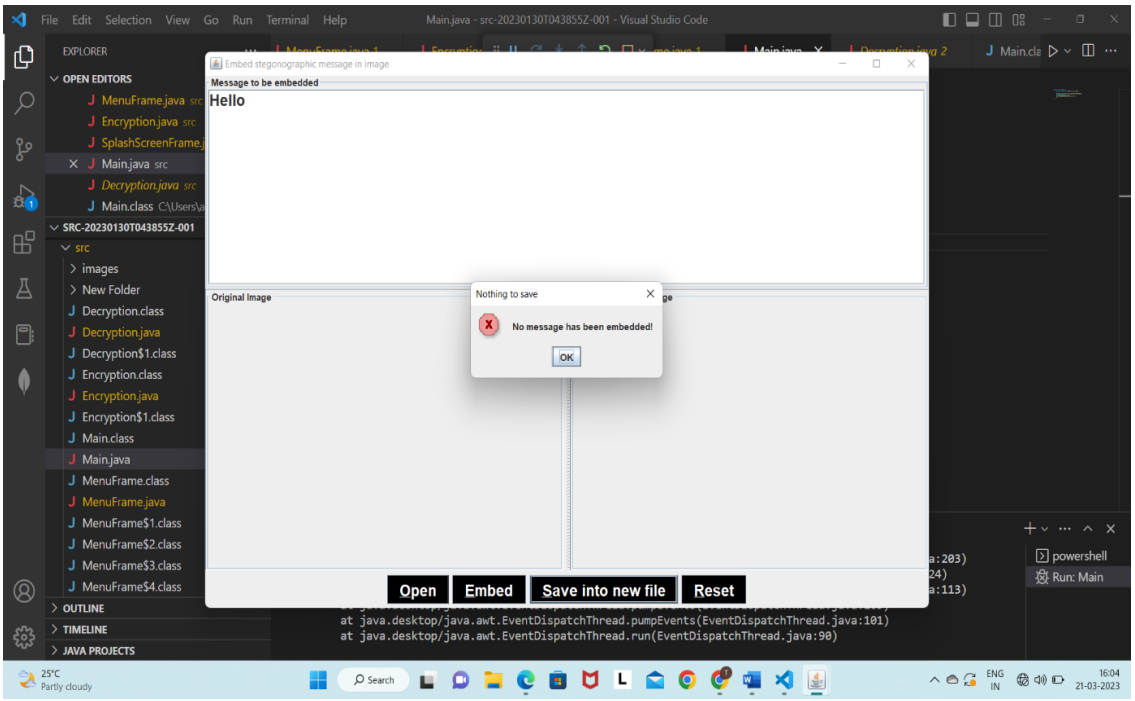

Figure 11.Embedding without selecting image

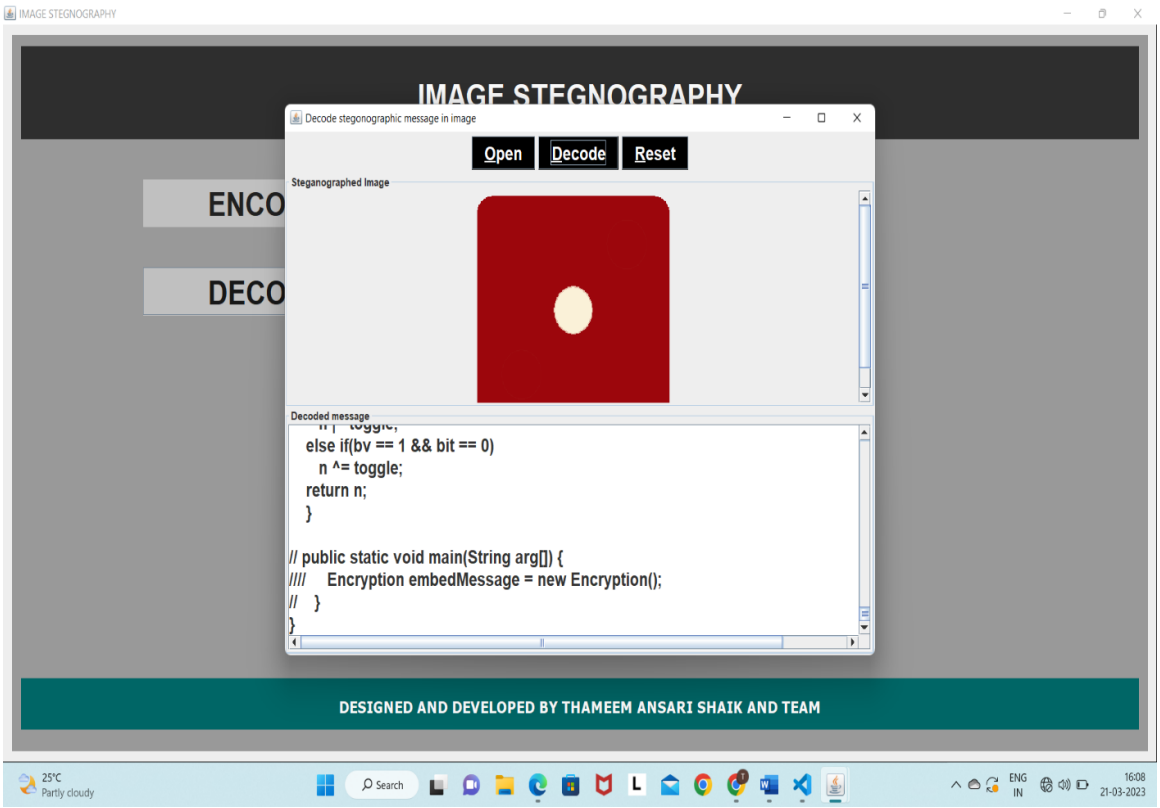

Figure 12.Decoding more lines of message

CHAPTER – 6

RESULTS & DISCUSSION

6.1 INTRODUCTION

To complete this study properly, it is necessary to discuss the achievements realized in relation to the objectives of the project so as to answer the research questions as well as limitations encountered during the project's timeline.

This chapter compasses the interpretation of the finding resulting from this study as well as the discussion on the development of the system.

6.2 OVERVIEW OF THE OUTPUT

The code that was executed and was stated in the previous chapter basically displays a main frame which prompts the user to select the operation he is willing to perform. If the user wishes to encrypt some information into an image, he/she can use the Encrypt button which redirects him to the encryption window. Similarly, if the user wishes to decrypt an image which he/she she received from the sender, one can choose the Decrypt button present in the main window which redirects him to the decryption window.

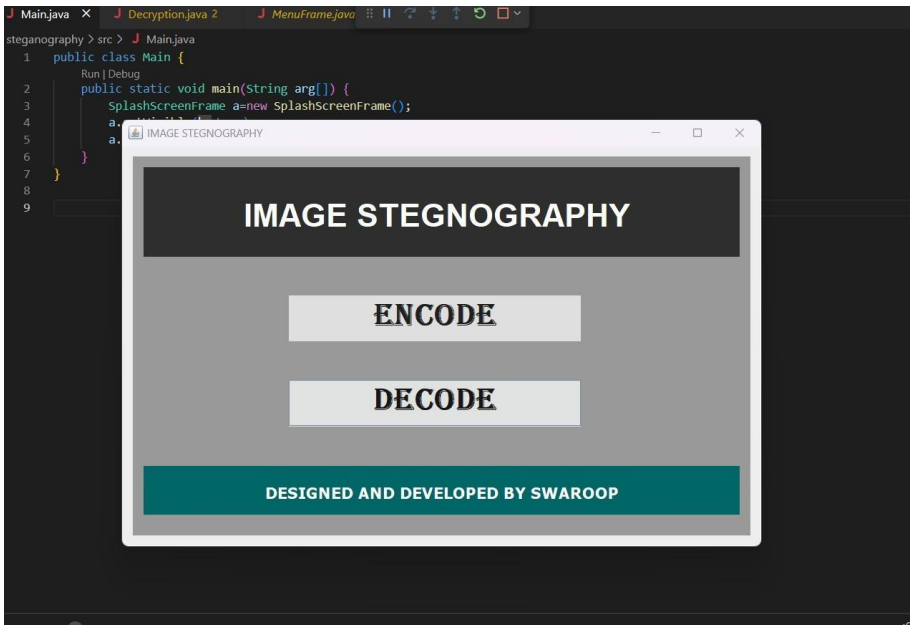

Figure 13. Main Output Window

6.2.1 ENCRYPTION WINDNOW

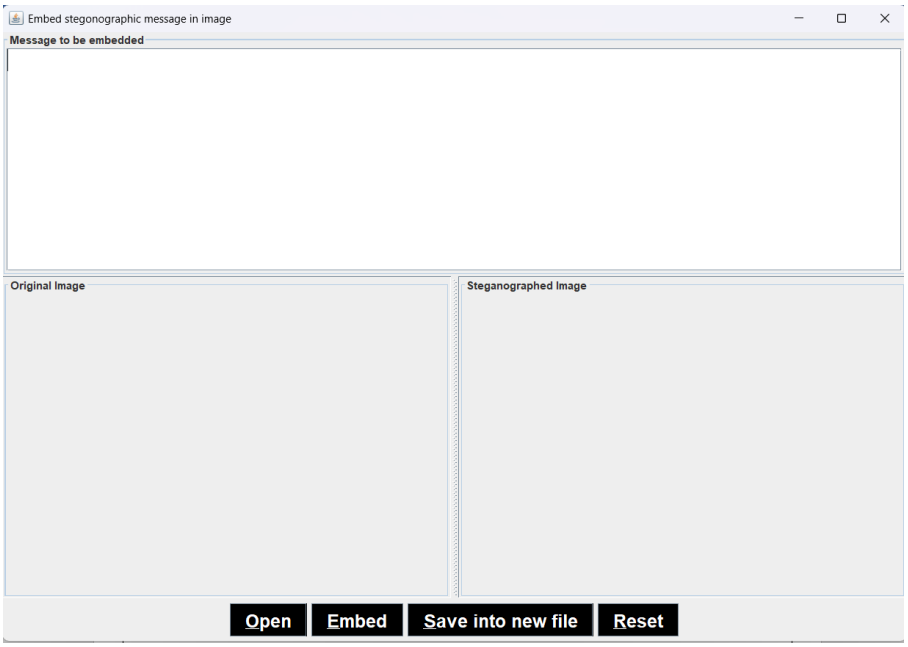

Figure 14. Basic Encryption window

Whatever the message the user wishes to encrypt into the image or the message that needs to be kept confidential, the message is typed in the upper portion of the encryption window and select the image form the local directory to embed the message.

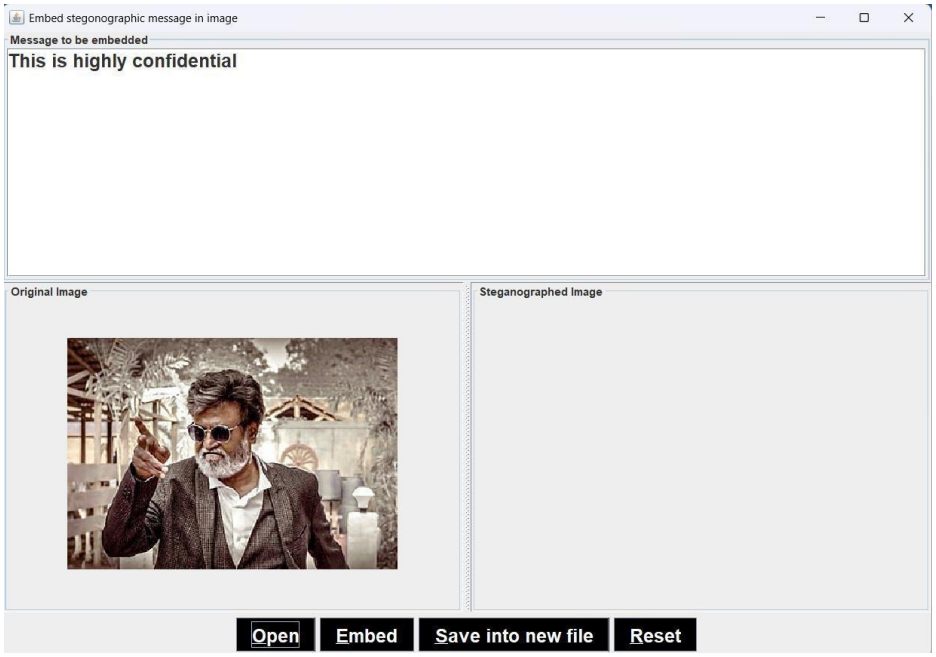

Figure 15. Select target image

As soon as the image is selected and the message is chosen, the steganographic image will be formed by hitting the Embed button present right beside the Open button which invokes the encryption file and results the steganographic image which is shown as,

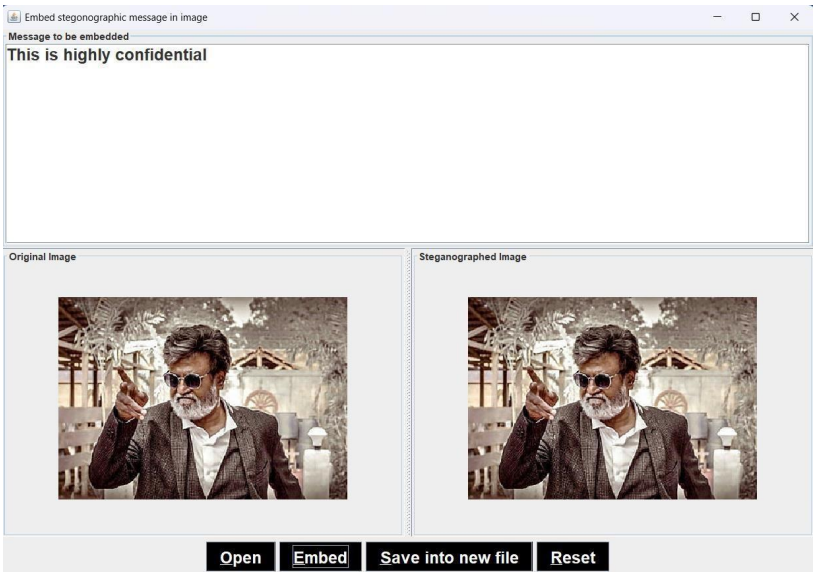

Figure 16. Embedded image

We can observe that the steganographic image is highly identical to the original image and we can save the resulted image by clicking the save into new file button which we can use to store the important information or we can share the image to the person we want to communicate secretly.

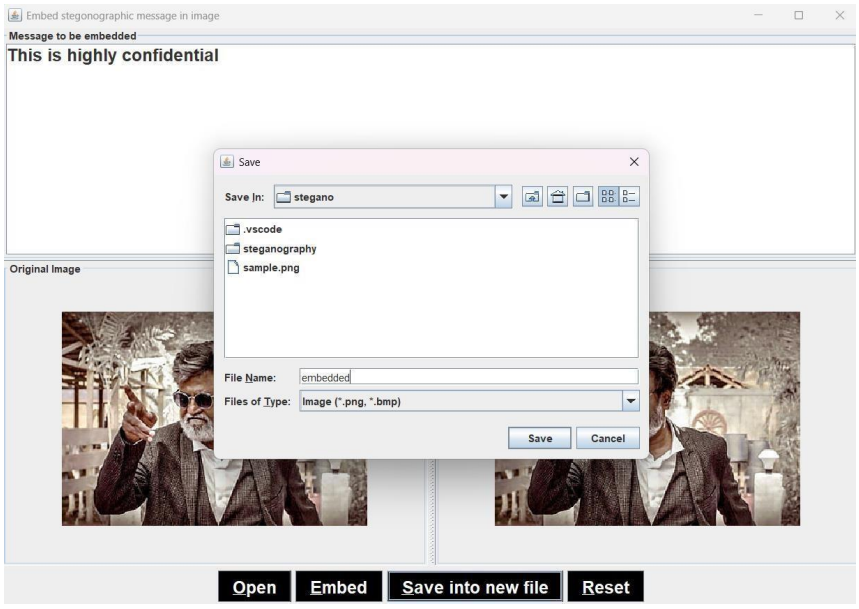

Figure 17. Save embed image

6.2.2 DECRYPTION WINDNOW

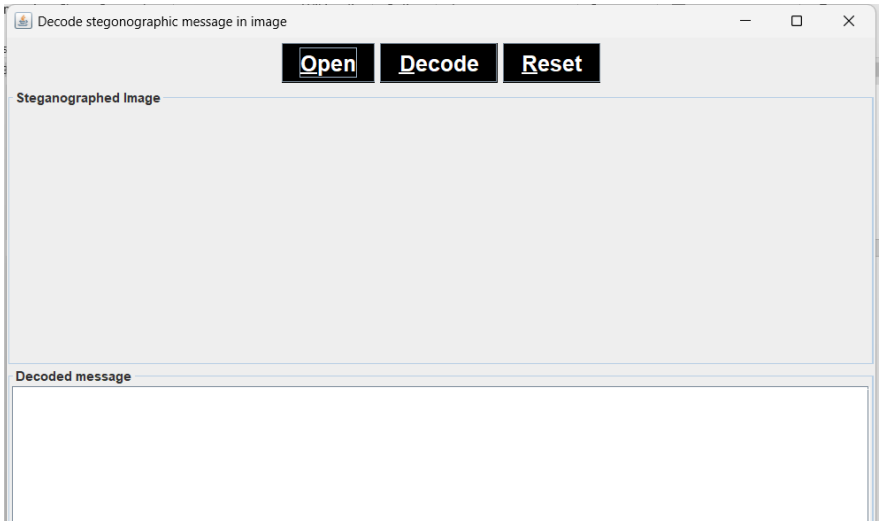

Figure 18. Basic Decryption window

As stated before, the decryption operation involves the extraction of embedded message from the image by selecting the embedded image in the decryption window through the Open Window.

The decrypted message will be displayed in the lower portion of the decryption window after clicking the Decode button.

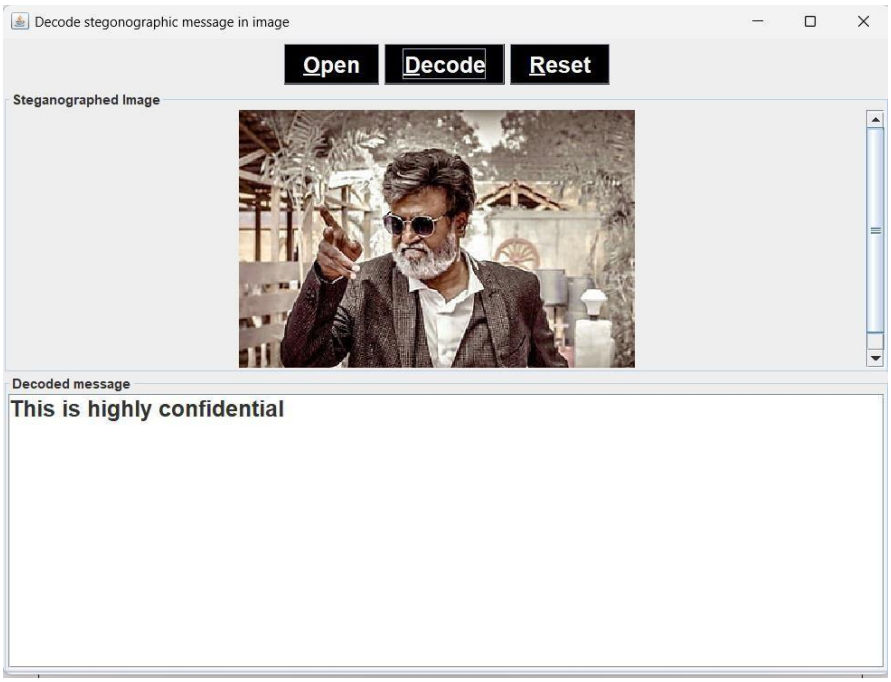

Figure 19. Final Decrypted output

CHAPTER - 7

CONCLUSION

7.1 INTRODUCTION

The purpose of this chapter is to summarize the thesis research and suggest research recommendations for further analysis. The first section of the chapter will give a summary of the research questions, followed by the finding to the research questions which answers the objectives of this paper The last part of the chapter will discuss the propose recommendations for further research.

7.2 CONCLUSION

On the basis of the problem statement in this study, this research came up with the following conclusions:

7.2.1 An encryption system which can hide information and data in an image file was the best ways to keep such data and information from unauthorized usage and modification. Since even a leaked stego-image will still require a known algorithm or the steganographic system to decrypt it. Moreover, it could be concluded that the implementation of the image steganographic system in the said company hides the very existence of the form of encryption deployed. This has helped to prevent suspecting the very existence of the hidden data or information file.

7.2.2 This means that the company has reduced the risk of information and data modification and access by unauthorized persons as well as making information in thecompany more manageable that is information is available, authentic, and confidentialand its integrity is protected. The stego-image resolution doesn't change much and is negligible when we embed the information or data file into the image.

7.2.3 I used the least significant bit algorithm in this project for developing the system because it is faster, reliable and compression ratio is moderate compared to other algorithms.

7.2.4 The major limitation of the system is that, the output file is limited a bitmap images (*.bmp) file. The system accepts only images as a carrier or cover file, and the compression depend on the document size as well as the carrier or cover image size.

7.2.5 And lastly, the study concludes that the personnel of the company are the fundamental barometers of the level of efficiency provided by the image steganographic system. In order to develop an ultimate Encryption system, a prior consultation among the employees would be advisable in order to cater the needs of who and which department will use the system and the types of images to use since every image has a copy right or an owner.

7.3 PROJECT SCOPE

This application would enable users to send confidential data of high priority to others. Because the human eye cannot decipher that there is any encrypted text, it can be put to great use while remaining easy to understand and use.

7.4 FUTURE SCOPE

In today's world, we often listen a popular term "Hacking". Hacking is nothing but an unauthorized access of data which can be collected at the time of data transmission. With respect to steganography this problem is often taken as Steganalysis. Steganalysis is a process in which an algorithm cracks the cover object to get the hidden data. So, whatever be the technique will be developed in future, degree of security related with that has to be kept in mind. It is hoped that Dual Steganography, Steganography along with Cryptography may be some of the future solution for this above-mentioned problem.

We hope to add support to hide all file formats. This allows for a much broader spectrum of uses: one would be able to encode .gif, .png, .pdf, .mp3, etc. The program would be more versatile because often hiding text just isn't enough. We also would like to implement batch image processing and statistical analysis so that we can run the program through a dataset of images and detect Steganography and perhaps crawl through Google Image Search to see how prevalent Steganography is.

8.REFERENCES

[1] Fridrich, Jessica, and Miroslav Goljan. "Digital image steganography using stochastic modulation." In *Security and Watermarking of Multimedia Contents V*, vol. 5020, pp. 191-202. SPIE, 2003.

[2] Johnson, N.F. and Jajodia, S., 1998. Exploring steganography: Seeing the unseen. *Computer*, *31*(2), pp.26-34.

[3] Provos, Niels, and Peter Honeyman. "Hide and seek: An introduction to steganography." *IEEE security & privacy* 1, no. 3 (2003): 32-44.

[4] Provos, Niels, and Peter Honeyman. *Detecting steganographic content on the internet*. Center for Information Technology Integration, 2001.

[5] Shengdong Zhang, Chunxiang Zhu, J. K. O. Sin and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," in IEEE Electron Device Letters, vol. 20, no. 11, pp. 569-571, Nov. 1999, doi: 10.1109/55.798046.

[6] Wegmuller, M., et al. "High resolution fiber distributed measurements with coherent OFDR." *Proc. ECOC'00*. Vol. 11. No. 4. 2000.

[7] Sorace, Ronald E., Victor S. Reinhardt, and Steven A. Vaughn. "High-speed digital-to- RF-converter." U.S. Patent 5,668,842, issued September 16, 1997.

[8] Pandit, A. S., Khope, S. R., & Student, F. Review on Image Steganography. International Journal of Engineering Science, 6115, (2016).

[9] A. Westfeld, "F5—a steganographic algorithm," in *Information Hiding*, I. S. Moskowitz, Ed., pp. 289–302, Springer, Berlin, Germany, 2001.

[10] Padhye, J., Firoiu, V., & Towsley, D. (1999). A stochastic model of tcp reno congestion avoidence and control.

[11] IEEE Computer Society LAN MAN Standard Committee. "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications." *IEEE Std. 802.11-1997* (1997).

[12] Organizing, Local. "IEEE INTERNATIONAL CONFERENCE ON PLASMA SCIENCE An Invitation to ICOPS 2004 In Baltimore, Maryland Visit our website at http://www. ieee. org/icops2004." (2004).

[13] Processor, FLEXChip Signal. "MC68175/D." (1996).

[14] Emam, M. M., Aly, A. A., &Omara, F. A. An Improved Image Steganography Method Based on LSB Technique with Random Pixel Selection. International Journal of Advanced Computer Science & Applications, 1(7), pp. 361-366, (2016).

[15] Al-Shatnawi, A. M. A new method in image steganography with improved image quality. Applied Mathematical Sciences, 6(79), 3907-3915, (2012).

[16] Artz, D. Digital steganography: hiding data within data. IEEE Internet computing, 5(3), 75-80, (2001).

[17] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285–287, 2006.