# A REAL TIME IMPLEMENTATION OF DATA HIDING IN AUDIO FOR MILITARY APPLICATIONS

*A Project Report*

*Submitted in partial fulfilment of Requirements for the Award of the Degree of*

## BACHELOR OF TECHNOLOGY
## IN
### ELECTRONICS AND COMMUNICATION ENGINEERING

***By***


| | |
|---|---|
| **CH. NAGA CHAITANYA** | **(218A1A0442)** |
| **A. NITHIN** | **(218A1A0438)** |
| **B. HARITH KUMAR REDDY** | **(218A1A0439)** |
| **CH. HEMANTH KUMAR REDDY** | **(218A1A0440)** |


***Under the Esteemed Guidance of***


**Miss. V. V. Narayanamma** **M. Tech**

Assistant Professor
Dept. of ECE




**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

RISE Krishna Sai Prakasam Group of Institutions::Ongole

(Approved by AICTE-New Delhi, Affiliated to JNTUK Kakinada)
(Accredited By NAAC with "A" Grade & NBA for B.Tech. in ECE, EEE, CE, ME and CSE)
An ISO 9001:2015 certified Institute
NH-16, Valluru, -523272, Ongole, Prakasam District, A.P


**2021- 2025**

# RISE Krishna Sai Prakasam Group of Institutions::Ongole

(Approved by AICTE-New Delhi, Affiliated to JNTUK Kakinada)
(Accredited By NAAC with "A" Grade & NBA for B.Tech. in ECE, EEE, CE, ME and CSE)
An ISO 9001:2015 certified Institute
NH-16, Valluru, -523272, Ongole, Prakasam District, A.P

**Department of Electronics & Communication Engineering**

***CERTIFICATE***

This is to certify that the project work entitled **" A REAL TIME IMPLEMENTATION OF DATA HIDING IN AUDIO FOR MILITARY APPLICATIONS"** is a bonafide record of project work done jointly by **CH.NAGA CHAITANYA (218A1A0442), A.NITHIN (218A1A0438), B.HARITH KUMAR REDDY (218A1A0439), CH.HEMANTH KUMMAR REDDY (218A1A0440),** under my guidance and supervision and is submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Electronics & Communication Engineering by Jawaharlal Nehru Technological University , Kakinada during the academic year 2024- 2025.

PROJECT GUIDE
**Miss .V.V.Narayanna** M.Tech
**Assistant Professor**
**Dept. of ECE**

HEAD OF THE DEPARTMENT
**Dr. Ch. Venugopal Reddy** M.E, Ph.D.
**Professor & HOD**
**Dept. of ECE**

EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our guide, to **Miss. V. V. Narayanamma** Assistant Professor, for providing his invaluable guidance, comments, suggestions, and support throughout the course of the project.

We express our heart filled thanks to **Dr. Ch. Venugopal Reddy**, Head of the Department, Electronics and Communication Engineering, for his valuable support to bring out this project in time.

We pay our profound sense of gratitude to our Principal **Prof. Dr. A.V. Bhaskara Rao** for providing an excellent environment in our college and helping us at all points for achieving our task.

We express our sincere thanks to the **Management of RISE Krishna Sai Prakasam Group of Institutions, Valluru** for providing a good environment and infrastructure.

Finally, we thank all our faculty members, supporting staff of ECE Department and friends for their kind co-operation and valuable help for the completion of the project.

**Project Associates**

| | |
|---|---|
| **CH. NAGA CHAITANYA** | **(218A1A0442)** |
| **A. NITHIN** | **(218A1A0438)** |
| **B. HARITH KUMAR REDDY** | **(218A1A0439)** |
| **CH. HEMANTH KUMAR REDDY** | **(218A1A0440)** |

| | |
|---|---|
| **Vision of the Institute** | To be a premier institution in technical education by creating professionals of global standards with ethics and social responsibility for the development of the nation and the mankind. |
| **Mission of the Institute** | Impart Outcome Based Education through well qualified and dedicated faculty. |
| | Provide state-of-the-art infrastructure and facilities for application-oriented research. |
| | Reinforce technical skills with life skills and entrepreneurship skills. |
| | Promote cutting-edge technologies to produce industry-ready Professionals. |
| | Facilitate interaction with all stakeholders of foster ideas and innovation. |
| | Inculcate moral values, professional ethics and social responsibility |
| **Vision of the Department** | To become a center of excellence in Electronics and Communication Engineering to meet the global, technological and industrial requirements. |
| **Mission of the Department** | Provide modern technical knowledge, professional skills and attitude to meet industry needs. |
| | Promote innovations through professional training and development. |
| | Develop a team with professional ethics and social responsibility. |

**Program Educational Objectives (PEOs):**

Graduates of the program will be able to

| | |
|---|---|
| **PEO1:** Core Skills | Intensive and extensive engineering knowledge and skill to understand, analyze, design and create novel products and solutions in the field of Signal Processing, Communication Systems, Embedded Systems and VLSI. |
| **PEO2:** Problem Solving and Lifelong Learning | Capability to pursue career in industry or higher studies with continuous learning. |
| **PEO3:** Entrepreneurship Skills | Leadership qualities, team spirit, multi-disciplinary approach, character Moulding and lifelong learning for a successful professional career. |
| **PEO4:** Professionalism | Professional and ethical attitude, effective communication skills, and sense of responsibility toward society. |

**Program Outcomes (POs):**

| | |
|---|---|
| **PO1** | **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| **PO2** | **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| **PO3** | **Design/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| **PO4** | **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |

| | |
|---|---|
| **PO5** | **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |
| **PO6** | **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. |
| **PO7** | **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **PO8** | **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| **PO9** | **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| **PO10** | **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 receive clear instructions. |
| **PO11** | **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. |
| **PO12** | **Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

**Program Specific Outcomes (PSOs):**

A student of the Electronics and Communication Engineering Program will be able to

| | |
|---|---|
| **PSO1** | Design and implementation of complex systems by applying basic concepts in Electronics & Communication Engineering to Electronics, Communications, Signal Processing, VLSI, Embedded Systems (**Core Skills**). |
| **PSO2** | Solve complex Electronics and Communication Engineering problems, using hardware and software tools, along with analytical skills to arrive cost effective and appropriate solutions relevant to the society (**Problem-Solving Skills**). |
| **PSO3** | Quality in technical subjects for successful higher studies and employment (**Professional Career**). |

# RISE Krishna Sai Prakasam Group of Institutions::Ongole

(Approved by AICTE-New Delhi, Affiliated to JNTUK Kakinada)
(Accredited By NAAC with "A" Grade & NBA for B.Tech. in ECE, EEE, CE, ME and CSE)
An ISO 9001:2015 certified Institute
NH-16, Valluru, -523272, Ongole, Prakasam District, A.P

## Project Outcomes

**Name of the Course** : Project Work **Year & Semester** : IV Year II Sem
**Academic Year** : 2024-2025 **Regulation** : R20

| Co. No | Project outcome | BTL |
|---|---|---|
| | After completing this project the student will be able to | |
| C421.1 | Envisaging applications for societal needs | Evaluating |
| C421.2 | Develops skills for analysis and synthesis of practical systems | Creating |
| C421.3 | Acquire the use of new tools effectively and creatively | Creating |
| C421.4 | Work in team to carry out analysis and cost-effective, environmental friendly designs of engineering systems | Creating |
| C421.5 | Write Technical / Project reports and oral presentation of the work done to an audience | Evaluating |
| C421.6 | Demonstrate a product developed | Creating |

# RISE Krishna Sai Prakasam Group of Institutions::Ongole

(Approved by AICTE-New Delhi, Affiliated to JNTUK Kakinada)
(Accredited By NAAC with "A" Grade & NBA for B.Tech. in ECE, EEE, CE, ME and CSE)
An ISO 9001:2015 certified Institute
NH-16, Valluru, -523272, Ongole, Prakasam District, A.P

**Name of the Course** : Project Work **Year & Semester** : IV Year II Sem
**Academic Year** : 2024-2025 **Regulation** : R20

**CO Vs PO Mapping**

| Course Outcomes (COs) | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C421.1 | 2 | 2 | 3 | 3 | 3 | 3 | 2 | 2 | 3 | 2 | 2 | 2 |
| C421.2 | 2 | 2 | 3 | 3 | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 3 |
| C432.3 | 2 | 2 | 3 | 3 | 3 | 3 | 2 | 2 | 3 | 2 | 2 | 2 |
| C421.4 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| C421.5 | 2 | 2 | 3 | 2 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 |
| C421.6 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| C421 | **2.00** | **2.00** | **2.83** | **2.67** | **3.00** | **3.00** | **2.33** | **2.50** | **3.00** | **2.67** | **2.67** | **2.67** |

**CO Vs PSO Mapping**

| Course Outcomes(COs) | PSO1 | PSO2 | PSO3 |
|---|---|---|---|
| C421.1 | 3 | 3 | 3 |
| C421.2 | 3 | 3 | 3 |
| C421.3 | 3 | 3 | 3 |
| C421.4 | 3 | 3 | 3 |
| C421.5 | 2 | 2 | 2 |
| C421.6 | 3 | 3 | 3 |
| C421 | **2.83** | **2.83** | **2.83** |

# CONTENTS

# LIST OF FIGURES

# ABSTRACT

In this paper we propose Secret communication through Audio with Textual Information using Steganography method. Steganography is an art of sending hidden data or secret messages over a public channel so that a third party cannot detect the presence of the secret messages. The goal of steganography is different from classical encryption, which seeks to conceal the content of secret messages; steganography is about hiding the very existence of the secret messages. Modern steganography is generally understood to deal with electronic media rather than physical objects. There have been numerous proposals for protocols to hide data in channels containing pictures, video, audio and even typeset text. This makes sense for a number of reasons.

# CHAPTER-1

## INTRODUCTION

### 1.1 Introduction

As the development of Internet technologies increases, the transmission of digital media is now-a-days convenient over the networks. But secret message transmissions over the Internet system suffer from serious security overhead. So, protecting of secret messages during transmission becomes an important issue. Though cryptography changes the message so that it cannot be understood but this can generates curiosity level of a hacker. It would be rather more sensible if the secret message is cleverly embedded in another media so that no one can guess if anything is hidden there or not. This idea results in steganography, which is a branch of information hiding by camouflaging secret information within other information. The word steganography in Greek means "covered writing" ( Greek words "stegos" meaning "cover" and "grafia" meaning "writing") [1]. The main objective of steganography is to hide a secret message inside harmless cover media in such a way that the secret message is not visible to the observer. Thus the stego_image should not diverge much from original cover image. In this generation, steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels.

The advantage of steganography, over cryptography alone, is that messages do not attract attention to themselves. Plainly visible encrypted messages—no matter how unbreakable—will arouse suspicion, and may in themselves be incriminating in countries where encryption is illegal. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. As a simple example, a sender might start with an innocuous image file and adjust the color of every 100th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

Among all digital file formats available nowadays image files are the most popular cover objects because they are easy to find and have higher degree of distortion tolerance over other types of files with high hiding capacity due to the redundancy of digital information representation of an image data.

## 1.2 Applications digital stegonography

Stegonography is not restricted to just hide information of the author in the work, there are various other purposes for which watermarking may be incorporated into an object. Some of them are

Ø Copyright Protection : For the protection of intellectual property, the data owner can embed a stegono representing the copyright information in his data.

Ø Fingerprinting: To trace the source of illegal copies, the owner can use a fingerprinting technique. This requires the owner to embed different information onto copied of the work provided to different customers. The information embedded can be a serial number, customer id etc.

Ø Data Authentication: Introducing fragile stego into the data can help ensure that the data is not processed or modified in anyway by the user.

Ø Indexing: Introducing watermarks in video mail, movies, news items can be used to index the data.

Ø Data Hiding: stego may be used to embed longer bits of information in the data. The earliest form of this is was in ancient Greece, where an author could hide his name in the text of the literary work. The term used to describe data hiding ,'LSB techinique'originated in Greece. This was also used by the Germans/Allies in WWII to send sensitive information to outposts by hiding it in postcards.

Ø Medical Safety: Stego containing the name of the patient can be embedded onto the X-Rays, MRI Scans & other test results help in instant identification of the result as belonging to a patient and thus avoid mix-ups which can lead to catastrophic consequences.

Ø Robustness: Robustness is a measure of the ability of the embedding algorithm to introduce the watermark in such a way that it is retained in the image despite several stages of image

processing. The image may be filtered ( high-pass or low-pass or median) rotated, translated, cropped , scaled etc. As part of image processing . A good watermarking algorithm embeds the stego in the spatial or frequency regions of the image, which would be least affected by such processing. Good correlation is possible between the recovered

Ø Security: Security of a stegotechnique can be judged the same way as with encryption techniques. Assuming that unauthorized parties know the algorithm used for the embedding, the security of the algorithm lies in the selection of key. Thus the algorithm is truly secure if knowing the exact algorithm to embed and extract data does not help an unauthorized party in actually recovering the data from the stego image.

Ø Payload of stego: The amount of information that can be stored in a stego depends on the application. For example, in copy protection purposes, a payload of one bit is more than sufficient. The latest proposal for audio stegonography standard specifies that an audio stegonography be at least 20 bits per second. (This is however almost impractical and so will be reduced to only a few bits ). For intellectual hide such as ISBN or ISRC a length of 60-70 bits would be sufficient. “stegonographygranularity” is a term used to refer to the number of bits that are actually needed to represent the entire stego in the image. Generally for video information the watermarking information can be spread over a few frames. Although this decreases the robustness, this approach still suffices for most applications.

## 1.3 Stegonography techniques

There are a number of steganographic schemes that hide secret message in an image file; these schemes can be classified according to the format of the cover image or the method of hiding. We have two popular types of hiding
Methods; spatial domain embedding and transform domain embedding.

1.visible stegonography

2.invisible stegonography

3.audio stegonography

4.image stegonography

5.video stegonography

Stegonography techniques over view:

The first recorded uses of steganography can be traced back to 440 BC when Herodotus mentions two examples of steganography in The Histories of Herodotus. Demaratus sent a warning about a forthcoming attack to Greece by writing it directly on the wooden backing of a wax tablet before applying its beeswax surface. Wax tablets were in common use then as reusable writing surfaces, sometimes used for shorthand. Another ancient example is that of Histiaeus, who shaved the head of his most trusted slave and tattooed a message on it. After his hair had grown the message was hidden. The purpose was to instigate a revolt against the Persians

### 1.3.1 PHYSICAL STEGANOGRAPHY

Steganography has been widely used, including in recent historical times and the present day. Possible permutations are endless and known examples include:

- Hidden messages within wax tablets: in ancient Greece, people wrote messages on the wood, then covered it with wax upon which an innocent covering message was written.
- Hidden messages on messenger's body: also used in ancient Greece. Herodotus tells the story of a message tattooed on a slave's shaved head, hidden by the growth of his hair, and exposed by shaving his head again. The message allegedly carried a warning to Greece about Persian invasion plans. This method has obvious drawbacks, such as delayed transmission while waiting for the slave's hair to grow, and the restrictions on the number and size of messages that can be encoded on one person's scalp.
- In WWII, the French Resistance sent some messages written on the backs of couriers using invisible ink.
- Hidden messages on paper written in secret inks, under other messages or on the blank parts of other messages.
- Messages written in Morse code on knitting yarn and then knitted into a piece of clothing worn by a courier.
- Messages written on the back of postage stamps.
- During and after World War II, espionage agents used photographically produced microdots to send information back and forth. Microdots were typically minute,

approximately less than the size of the period produced by a typewriter. WWII microdots needed to be embedded in the paper and covered with an adhesive (such as collodion). This was reflective and thus detectable by viewing against glancing light. Alternative techniques included inserting microdots into slits cut into the edge of post cards.

- During World War II, a spy for Japan in New York City, Velvalee Dickinson, sent information to accommodation addresses in neutral South America. She was a dealer in dolls, and her letters discussed how many of this or that doll to ship. The stegotext was the doll orders, while the concealed "plaintext" was itself encoded and gave information about ship movements, etc. Her case became somewhat famous and she became known as the Doll Woman.
- Cold War counter-propaganda. In 1968, crew members of the USS Pueblo (AGER-2) intelligence ship held as prisoners by North Korea, communicated in sign language during staged photo opportunities, informing the United States they were not defectors but rather were being held captive by the North Koreans. In other photos presented to the US, crew members gave "the finger" to the unsuspecting North Koreans, in an attempt to discredit photos that showed them smiling and comfortable.

### 1.3.2 DIGITAL STEGANOGRAPHY

Modern steganography entered the world in 1985 with the advent of the personal computer being applied to classical steganography problems. Development following that was slow, but has since taken off, going by the number of "stego" programs available: Over 800 digital steganography applications have been identified by the Steganography Analysis and Research Center. Digital steganography techniques include:

- Concealing messages within the lowest bits of noisy images or sound files.
- Concealing data within encrypted data or within random data. The data to be concealed is first encrypted before being used to overwrite part of a much larger block of encrypted data or a block of random data (an unbreakable cipher like the one-time pad generates ciphertexts that look perfectly random if you don't have the private key).
- Chaffing and winnowing.
- Mimic functions convert one file to have the statistical profile of another. This can thwart statistical methods that help brute-force attacks identify the right solution in a ciphertext-only attack.

- Concealed messages in tampered executable files, exploiting redundancy in the targeted instruction set.
- Pictures embedded in video material (optionally played at slower or faster speed).
- Injecting imperceptible delays to packets sent over the network from the keyboard. Delays in keypresses in some applications (telnet or remote desktop software) can mean a delay in packets, and the delays in the packets can be used to encode data.
- Changing the order of elements in a set.
- Content-Aware Steganography hides information in the semantics a human user assigns to a datagram. These systems offer security against a non-human adversary/warden.
- Blog-Steganography. Messages are fractionalized and the (encrypted) pieces are added as comments of orphaned web-logs (or pin boards on social network platforms). In this case the selection of blogs is the symmetric key that sender and recipient are using; the carrier of the hidden message is the whole blogosphere.
- Steganography can be applied to different types of media including text, audio, image and video etc. However, text steganography is considered to be the most difficult kind of steganography due to lack of redundancy in text as compared to image or audio but still has smaller memory occupation and simpler communication. The method that could be used for text steganography is data compression. Data compression encodes information in one representation into another representation. The new representation of data is smaller in size. One of the possible schemes to achieve data compression is Huffman coding. Huffman coding assigns smaller length code words to more frequently occurring source symbols and longer length code words to less frequently occurring source symbols.

# CHAPTER-2

# INTRODUCTION TO SPEECH PROCESSING

## 2.1 Introduction to speech processing:

Since even before the time of Alexander Graham Bell's revolutionary invention, engineers and scientists have studied the phenomenon of speech communication with an eye on creating more efficient and effective systems of human-to-human and human-to-machine communication. Starting in the 1960s, digital signal processing (DSP), assumed a central role in speech studies, and today DSP is the key to realizing the fruits of the knowledge that has been gained through decades of research. Concomitant advances in integrated circuit technology and computer architecture have aligned to create a technological environment with virtually limitless opportunities for innovation in speech communication applications. In this chapter We present a comprehensive overview of digital speech processing that ranges from the basic nature of the speech signal. hence our goal is to provide a useful introduction to the wide range of important concepts that comprise the field of digital speech processing.

The fundamental purpose of speech is communication, i.e., the transmission of messages. According to information theory, a message represented as a sequence of discrete symbols can be quantified by its *information content* in bits, and the rate of transmission of information is measured in bits/second (bps). In speech production, as well as in many human-engineered electronic communication systems, the information to be transmitted is encoded in the form of a continuously varying (analog) waveform that can be transmitted, recorded, manipulated, and ultimately de coded by a human listener.

In the case of speech, the fundamental analog form of the message is an acoustic waveform, which we call the *speech signal*. Speech signals, as illustrated in Figure 1.1, can be converted to an electrical waveform by a microphone, further manipulated by both analog and digital signal processing, and then converted back to acoustic form by a loudspeaker, a telephone handset or headphone, as desired. This form of speech processing is, of course, the basis for Bell's telephone invention as well as today's multitude of devices for recording, transmitting, and manipulating speech and audio signals.

Although Bell made his invention without knowing the fundamentals of information theory, these ideas have assumed great importance in the design of sophisticated modern communications systems. Therefore, even though our main focus will be mostly on the speech waveform and its representation in the form of parametric models, it is nevertheless useful to begin with a discussion of how information is encoded in the speech waveform.

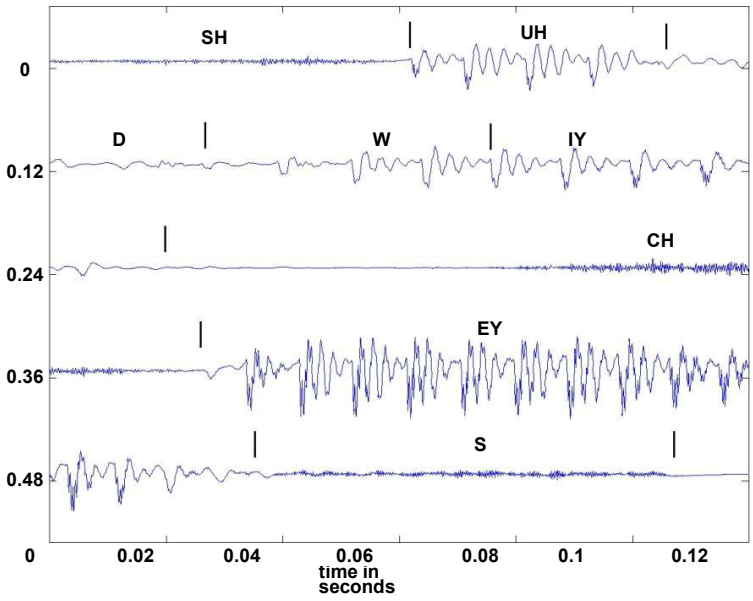


**Fig. 2.1: A speech waveform with phonetic labels for the text message "Should we chase."**

The purpose of speech is communication. There are several ways of characterizing the communications potential of speech. One highly quantitative approach is in terms of information theory ideas as according to information theory, speech can be represented in terms of its message content, or information. An alternative way of characterizing speech is in terms of the signal carrying the message information, i.e., the acoustic waveform.

## 2.2 Digital Speech Processing

**Speech processing** is the study of speech signals and the processing methods of these signals.

The signals are usually processed in a digital representation, so speech processing can be regarded as a special case of digital signal processing, applied to speech signal.

Speech processing can be divided into the following categories:

- **Speech recognition**, which deals with analysis of the linguistic content of a speech signal.
- **Speaker recognition**, where the aim is to recognize the identity of the speaker.

- **Speech coding**, a specialized form of data compression, is important in the telecommunication area.

- **Voice analysis,** for medical purposes, such as analysis of vocal loading and dysfunction of the vocal cords.

- **Speech synthesis:** the artificial synthesis of speech, which usually means computer-generated speech.

- **Speech enhancement**: enhancing the intelligibility and/or perceptual quality of a speech signal, like audio noise reduction for audio signals.

The first step in most applications of digital speech processing is to convert the acoustic waveform to a sequence of numbers. Most modern A-to-D converters operate by sampling at a very high rate, applying a digital low pass filter with cut of set to preserve a prescribed bandwidth, and then reducing the sampling rate to the desired sampling rate, which can be as low as twice the cut of frequency of the sharp-cutoff digital filter.

This discrete-time representation is the starting point for most applications. From this point, other representations are obtained by digital processing. For the most part, these alternative representations are based on incorporating knowledge about the workings of the speech chain. As we will see, it is possible to incorporate aspects of both the speech production and speech perception process into the digital representation and processing. It is not an over-simplification to assert that digital speech processing is grounded in a set of techniques that have the goal of pushing the data rate of the speech representation to the left along either the upper or lower path.

## 2.3 Speech Coding

Perhaps the most widespread applications of digital speech processing technology occur in the areas of digital transmission and storage of speech signals. In these areas the centrality of the digital representation is obvious, since the goal is to compress the digital wave-form representation of speech into a lower bit-rate representation. It is common to refer to this activity as “speech coding” or “speech compression.

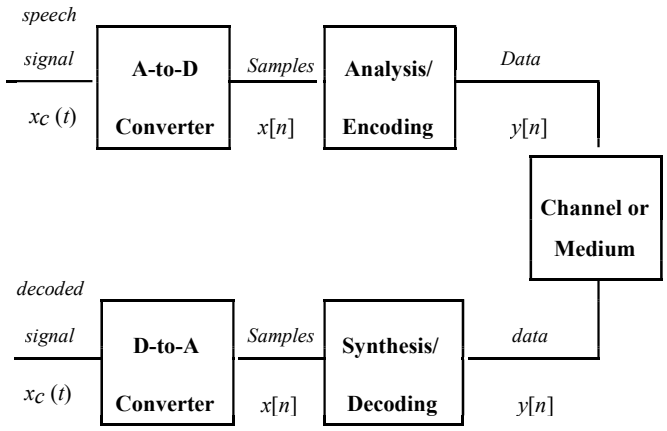


**Fig.2.2 : Speech coding block diagram — encoder and decoder**

Figure 1.2 shows a block diagram of a generic speech encoding/decoding (or compression) system. In the upper part of the figure, the A-to-D converter converts the analog speech signal $x_c(t)$ to a sampled waveform representation $x[n]$. The digital signal $x[n]$ is analyzed and coded by digital computation algorithms to produce a new digital signal $y[n]$ that can be transmitted over a digital communication channel or stored in a digital storage medium as $y\hat{}[n]$. As we will see, there are a myriad of ways to do the encoding so as to reduce the data rate over that of the sampled and quantized speech waveform $x[n]$.

Because the digital representation at this point is often not directly related to the sampled speech waveform, $y[n]$ and $y\hat{}[n]$ are appropriately referred to as *data signals* that represent the speech signal. The lower path in Figure 1.2 shows the decoder associated with the speech coder. The received data signal $y\hat{}[n]$ is decoded using the inverse of the analysis processing, giving the sequence of samples $x\hat{}[n]$ which is then converted (using a D-to-A Converter) back to an analog signal $x\hat{}_c(t)$ for human listening. The decoder is often called a *synthesizer* because it must reconstitute the speech waveform from data that may bear no direct relationship to the waveform.

With carefully designed error protection coding of the digital representation, the transmitted ($y[n]$) and received ($\hat{}y[n]$) data can be essentially identical. This is the quiet essential feature of digital coding. In theory, perfect transmission of the coded digital representation is possible even under very noisy channel conditions, and in the case of digital storage, it is possible to store a perfect copy of the digital representation in perpetuity if sufficient care is taken to update the storage medium as storage technology advances.

This means that the speech signal can be reconstructed to within the accuracy of the original coding for as long as the digital representation is retained. In either case, the goal of the speech coder is to start with samples of the speech signal and reduce (compress) the data rate required to represent the speech signal while maintaining a desired perceptual fidelity. The compressed representation can be more efficiently transmitted or stored, or the bits saved can

be devoted to error protection.

Speech coders enable a broad range of applications including narrowband and broadband wired telephony, cellular communications, voice over internet protocol (VoIP) (which utilizes the internet as a real-time communications medium), secure voice for privacy and encryption (for national security applications), extremely narrowband communications channels (such as battlefield applications using high frequency (HF) radio), and for storage of speech for telephone answering machines, interactive voice response (IVR) systems, and pre-recorded messages. Speech coders often utilize many aspects of both the speech production and speech perception processes, and hence may not be useful for more general audio signals such as music.

Coders that are based on incorporating only aspects of sound perception generally do not achieve as much compression as those based on speech production, but they are more general and can be used for all types of audio signals. These coders are widely deployed in MP3 and AAC players and for audio in digital television systems.

## 2.4 Speech Recognition

Another large class of digital speech processing applications is concerned with the automatic extraction of information from the speech signal. Most such systems involve some sort of pattern matching. Figure 1.3 shows a block diagram of a generic approach to pattern matching problems in speech processing. Such problems include the following: speech recognition, where the object is to extract the message from the speech signal; speaker recognition, where the goal is to identify who is speaking; speaker verification, where the goal is to verify a speaker's claimed identity from analysis of their speech


**Fig. 2.3 : Block diagram of general pattern matching system for speech signals.**

signal; word spotting, which involves monitoring a speech signal for the occurrence of specified words or phrases; and automatic indexing of speech recordings based on recognition (or spotting) of spoken keywords.

The first block in the pattern matching system converts the analog speech waveform to digital form using an A-to-D converter. The feature analysis module converts the sampled speech signal to a set of feature vectors. Often, the same analysis techniques that are used in speech coding are also used to derive the feature vectors. The final block in the system, namely the pattern matching block, dynamically time aligns the set of feature vectors representing the speech signal with a concatenated set of stored patterns, and chooses the identity associated with the pattern which is the closest match to the time-aligned set of feature vectors of the

speech signal. The symbolic output consists of a set of recognized words, in the case of speech recognition, or the identity of the best matching talker, in the case of speaker recognition, or a decision as to whether to accept or reject the identity claim of a speaker in the case of speaker verification.

Although the block diagram of Figure 1.3 represents a wide range of speech pattern matching problems, the biggest use has been in the area of recognition and understanding of speech in support of human– machine communication by voice. The major areas where such a system finds applications include command and control of computer software, voice dictation to create letters, memos, and other documents, natural language voice dialogues with machines to enable help desks and call centers, and for agent services such as calendar entry and update, address list modification and entry, etc.

Pattern recognition applications often occur in conjunction with other digital speech processing applications. For example, one of the preeminent uses of speech technology is in portable communication devices. Speech coding at bit rates on the order of 8 Kbps enables normal voice conversations in cell phones. Spoken name speech recognition in cell phones enables voice dialing capability that can automatically dial the number associated with the recognized name. Names from directories with upwards of several hundred names can readily be recognized and dialed using simple speech recognition technology.

Another major speech application that has long been a dream of speech researchers is *automatic language translation.* The goal of language translation systems is to convert spoken words in one language to spoken words in another language so as to facilitate natural language voice dialogues between people speaking different languages. Language translation technology requires speech synthesis systems that work in both languages, along with speech recognition (and generally natural language understanding) that also works for both languages; hence it is a very difficult task and one for which only limited progress has been made. When such systems exist, it will be possible for people speaking different languages to communicate at data rates on the order of that of printed text reading!

## 2.5 Speech Processing Applications

The range of speech communication applications is illustrated in Figure 1.4. As seen in this figure, the techniques of digital speech processing are a key ingredient of a wide range of applications that include the three areas of transmission/storage, speech synthesis, and speech recognition as well as many others such as speaker identification, speech signal quality enhancement, and aids for the hearing- or visually-impaired.

The block diagram in Figure 1.5 represents any system where time signals such as speech are processed by the techniques of DSP. This figure simply depicts the notion that once the speech signal is sampled, it can be manipulated in virtually limitless ways by DSP techniques.

Here again, manipulations and modifications of the speech signal are

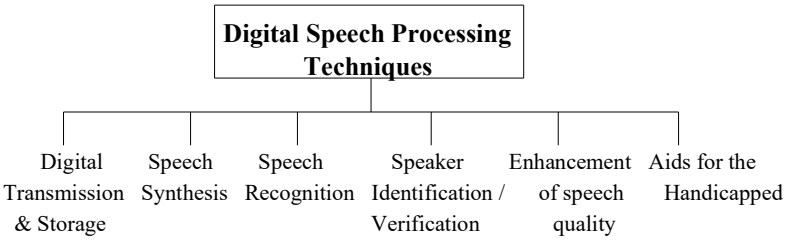


**Fig. 2.4: Range of speech communication applications**

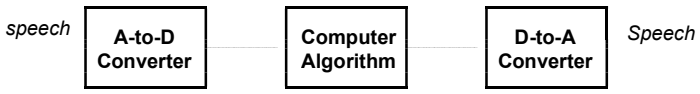


Fig. 2.5 : General block diagram for application of digital signal processing to speech signals.

usually achieved by transforming the speech signal into an alternative representation (that is motivated by our understanding of speech production and speech perception), operating on that representation by further digital computation, and then transforming back to the wave-form domain, using a D-to-A converter.

One important application area is *speech enhancement*, where the goal is to remove or suppress noise or echo or reverberation picked up by a microphone along with the desired speech signal. In human-to-human communication, the goal of speech enhancement systems is to make the speech more intelligible and more natural; however, in reality the best that has been achieved so far is less perceptually annoying speech that essentially maintains, but does not improve, the intelligibility of the noisy speech. Success *has* been achieved, however, in making distorted speech signals more useful for further processing as part of a speech coder, synthesizer, or recognizer.

# CHAPTER-3

# LITERATURE SURVEY

## 3.1 DATA ENCRYPTING IN A BINARY IMAGE BASE ON MODIFIED DATA HIDING METHOD

This encryption method manipulates sub-divided blocks using modified bit position to replace a secret bit. The sub-divided block contains three or more pixels of the host binary image. For every block decides to hide a secret bit. By finding the pixel position to insert a secret bit for each block, the image quality of the overt binary image can be improved.

## 3.2 ENCRYPTING PROCESSES

### 3.2.1 MODIFIED DATA HIDING METHOD

Let H be the host binary image of MXN pixels and C be the mXn-bit secret data. For pixel value h(i,j) of H, a new pixel value is defined as h'(i,j). The following processes are executed to hide a data bit.

1. For a given H. Select a sub-divided block Buv with size pXq for hiding a secret data bit.
2. Summing all pixels of Buv.
3. If **S**(**B**uv) is equal to 0 or pXq, is not used to store a secret data bit in the block.
4. If mod(**S**(**B**uv),2) is equal to the **C**(z), then do not make a change and save the data bit in this block.
5. If mod(**S**(**B**uv),2) is not equal to the **C**(z),

### 3.2.2 ENCRYPTING PROCESSES

Let **H** be a host binary image and let **H*** be an overt binary image modified from **H**. The elements of the overt image **H*** contain encrypted codes and the codes are classified into five groups of codes. The identification codes are used to determine if the codes encrypted in **H*** use the encrypted method proposed in this paper or not; the initial position codes are used to assign the initial position of the top-left of the sub-divided block; the sub-divided block dimension codes are used to indicate the size of the sub-divided block; the covert binary image

dimension codes are used to indicate the size of the covert binary image; (all the above-mentioned codes locate at the first row to the second one of **H***) the information codes are used to decrypt covert binary image (locate at the third row to the last one of **H***). Identification codes are like passwords and they are used to determine whether an overt image contains codes proposed in this paper or not. Identification codes are composed of 20-bit of pseudo-random binary codes, e.g. 10011000010000100001. Initial position codes are used to assign the initial position of the top-left of the sub-divided block and they need two sets of 4-bit binary codes. The first set shows the number of the row position and the second set shows the number of the column position. The codes 0000, 0011, 0111, and 1111 are used to represent number of 1, 4, 8, and 16, respectively. Other number codes can be used similarly.

**MERITS**

- PSNR value was high.
- Easy to encrypt binary image.

**DEMERITS**

- If there was any change in the binary information, we cannot reconstruct the encrypted data

## 3.3 VISUAL CRYPTOGRAPHIC STEGANOGRAPHY IN IMAGES

Cryptography involves converting a message text into an unreadable cipher. On the other hand, steganography embeds message into a cover media and hides its existence. Both these techniques provide some security of data neither of them alone is secure enough for sharing information over an unsecure communication channel and are vulnerable to intruder attacks. In this paper we propose an advanced system of encrypting data that combines the features of cryptography, steganography along with multimedia data hiding. This system will be more secure than any other these techniques alone and also as compared to steganography and cryptography combined systems.

**1. Encryption Algorithm:** The message will first be encrypted using Asymmetric Key Cryptography technique. The data will be encrypted using basic DES algorithm. This cipher will now be hidden into a multimedia file. The cipher will be saved in the image using a modified bit encoding technique by truncating the pixel values to the nearest zero digit (or a

predefined digit) and then a specific number which defines the 3-D representation of the character in the cipher code sequence can be added to this number. For every character in the message a specific change will be made in the RGB values of a pixel. (This change should be less than 5 for each of R, G and B values) This deviation from the original value will be unique for each character of the message. This deviation also depends on the specific data block (grid) selected from the reference database. For each byte in the data one pixel will be edited. Thus one byte of data will be stored per pixel in the image. In this method the cipher sequence can be decoded without the original image and only the edited image will be transmitted to the receiver. In the first few lines of image properties, the attributes of the image will be encrypted and saved so as to provide us the information if the image is edited or modified or the image extension has been changed like jpg to gif. These properties can be used in the decoding (identifying the correct block of data from the data grid). So only the correct encrypted image in the correct format will produce the sent message. For decryption, the receiver must know which image to decode and in which format as changing the image format changes the color distribution of the image. Every image gives a random data on decryption that has no meaning. But only the correct format decryption gives the original message. After hiding the data in the image, the image will be sent to the receiver. The receiver should have the decryption key (private key) which will be used to decode the data.

**2. Decryption Algorithm:** The message can be decoded using an inverse function (as used in traditional techniques) using the receiver's private key. This key can be a part of the image or a text or any attribute of the image. The receiver's private key is used to identify the reference grid from the reference database. After selecting the correct grid, the x and y component of the image can define the block that has been used to encrypt the message and the RGB values can point to the data in the block identified by the x, y component. The cipher is retrieved by obtaining the difference in the pixel value from the closest predefined value (zero truncation). These numbers will now define the saved bit and will form the cipher text. This cipher can now be decrypted using an inverse function of the DEA algorithm to get the message text.

**MERITS**

- We can hide text data in any image.
- Easy to handle.

**DEMERITS**

- Easy to hack.
- Computational complexity was high.

## 3.4 IMAGE STEGANOGRAPHY USING MOD-4 EMBEDDING ALGORITHM BASED ON IMAGE CONTRAST

In order to improve the capacity of the hidden secret data and to provide an imperceptible stego image quality, a new image steganography method based on image contrast is presented. A group of 2×2 blocks of non-overlapping spatially adjacent pixels is selected as the valid block for embedding the secret message. The modulo 4 arithmetic operation is further applied to all the valid blocks to embed a pair of binary bits using the shortest route modification scheme. Each secret message is also encrypted by RSA encryption algorithm to provide the system with more security.

Encryption: In this section we propose a RSA public key encryption for encrypting the secret message before embedded into cover image. RSA can be used for both encryption and decryption. In public key encryption the sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message. draw.

**Data Hiding:** In this section we propose a mod-4 embedding method for information hiding within the spatial domain of any gray scale image. This method can be considered as the improved version of. The input messages can be in any digital form, and are often treated as a bit stream. Embedding pixels are selected based on some mathematical function which depends on the pixel intensity value of the valid blocks in an image. Before embedding a checking has been done to find out whether the selected embedding pixels lies at the boundary of the image or not. Data embedding are done by mapping each two bits of the secret message in each of the valid block based on some features of that pixel.

**Data Hiding Model:** The input messages can be in any digital form, and are often treated as a bit stream. The input message is first converted into encrypted form through proposed encryption method. This encrypted message generates the secret key which may be used as a password before starting of the embedding or extracting operation for increasing another level

of security. Second the image is reshaped to the 2×2 blocks of non-overlapping spatially adjacent pixels. Then the valid blocks are selected from these blocks. Block Q is valid if the average difference between the gray level values of the pixels of that and it's mean (C) exceeds a threshold (minimum contrast)

**MERITS**

- Image contrast was enhanced and data was hidden.
- We cannot identify the given image was data hidden image.

**DEMERITS**

- For improving contrast, the pixel values in the given image were changed. This makes change in the data.
- Data bits in the image can change.

## 3.5 IMPLEMENTATION AND ANALYSIS OF THREE STEGANOGRAPHIC APPROACHES

This paper proposes the enhance security system by combining these two techniques. In this system, the encrypted message is embedded in a BMP image file. In proposed system, three LSB steganographic techniques have been implemented and analyzed. This proposed system intends for data confidentiality, data authentication and data integrity. This system not only enhances the security of data but also becomes more powerful mechanism. This system intends to support effective ways for protecting data. The primary goal of our system is to enhance the security of data and then to compare three steganographic techniques. Then we will use the optimized method for embedding. In this paper, we just present three steganographic approaches. In this system, data is encrypted with RC4 encryption algorithm and then embedded the encrypted text in the BMP image file using three steganographic methods.

**RC4 Encryption Algorithm:** In this paper, we use RC4 encryption algorithm. It is a variable key size cipher and symmetric key algorithm. Variable key size is from 1 to 256 bit to initialize a 256 bit state table. State table is used for subsequent generation of pseudo-random bits and then to generate a pseudo-random stream. The algorithm has two stages: initialization and operation.

**RIPEMD-160 hashing function:** Hash algorithms are important components in many cryptographic applications and security protocol suites. Hah functions, also called message digests and one way encryption, use no key. They are also employed by many operating systems to encrypt passwords. Therefore, it provides a measure of the integrity of a file. In this paper, we use RIPEMD-160 hash algorithms to provide higher protection. RIPEMD-160 has been designed by Hans Dobbertin, Antoon Bosselaers and Bart Preneel and produces a 160 bit output after performing five independent rounds. Each round is composed of 16 iterations resulting in 80 iterations in total.

**MERITS**

- This system uses cryptography and steganography to enhance the security. By combining these two techniques, it can enhance confidentiality and integrity of information.

**DEMERITS**

- Computation cost was high.
- Time complexity was high.

## 3.6 REVERSIBLE DATA HIDING IN ENCRYPTED IMAGE

This work proposes a novel reversible data hiding scheme for encrypted image. After encrypting the entire data of an uncompressed image by a stream cipher, the additional data can be embedded into the image by modifying a small proportion of encrypted data. With an encrypted image containing additional data, one may firstly decrypt it using the encryption key, and the decrypted version is similar to the original image.

A content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image, and then a data-hider embeds additional data into the encrypted image using a data-hiding key though he does not know the original content. With an encrypted image containing additional data, a receiver may firstly decrypt it using the encryption key, and the decrypted version is similar to the original image. According to the data-hiding key, he can further extract the embedded data and recover the original image from the decrypted version.

**Image Encryption:** Assume the original image is in uncompressed format and each pixel with gray value falling into [0, 255] is represented by 8 bits. Denote the bits of a pixel as $b_{i,j,0}, b_{i,j,1}, \ldots, b_{i,j,7}$ where indicates the pixel position, and the gray value as $p_{i,j}$.

**Data Embedding:** With the encrypted data, although a data-hider does not know the original image content, he can embed additional message into the image by modifying a small proportion of encrypted data.

**Data Extraction and Image Recovery:** When having an encrypted image containing embedded data, a receiver firstly generates $r_{i,j,k}$ according to the encryption key, and calculates the exclusive-or of the received data and $r_{i,j,k}$ to decrypt the image. We denote the decrypted bits as $b_{i,j,k}$.

**MERITS**

- High security.
- Computational complexity was low.

**DEMERITS**

- Since there was no separate key for data decode and decryption, any person can retrieve data and image if he got encrypted key.

# CHAPTER 4

## EXISTING SYSTEM

### 4.1 Cryptography:

Cryptography is an art of protecting the information by transforming into an unreadable and untraceable format known as cipher text. Only the person who possess the secret key can decipher or we can say decrypt the message into the original form. Cryptography is the technique by which one can send and share the information in a secret manner. Due the cryptography the information seems to be appearing like a garbage value and it is always almost impossible to find the information content lying under the image or a video file. The information looks like hidden inside the image or the video file. A very simplest and well known algorithm for cryptography is as shown in fig 2. The encryption key generator is used to generate the encryption key as well as the public key as shown in the below block diagram. By using the encryption key the information content to be sent gets encrypted by the encryptor. The encrypted information is then transmitted to the particular receiver. At the receiver end the cryptography decryptor is used which extracts the original information content mapped onto the image or a video file with the help of a public key provided by the transmitter section. By the use of the cryptography method only, the receiver which has the knowledge of the public key can retrieve the original information content from the image or a video file. So even if any unwanted person or a source gets the image or a video file with information content hidden in it, it cannot be extracted without proper public key. So public key plays a vital role in the whole cryptography process.

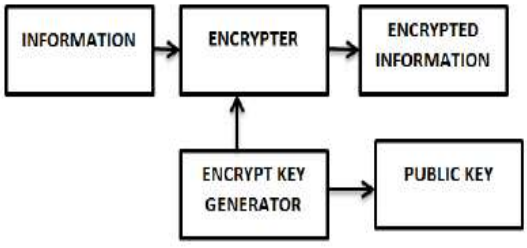


Fig.4.1. Cryptography encrypter

Technically in simple words “cryptography means hiding one piece of data within another”. Modern cryptography uses the opportunity of hiding information into digital multimedia files and also at the network packet level. Hiding information into a media requires following elements. The cover media(C) that will hold the hidden data

- The secret message (M), may be plain text, cipher text or any type of data
- The stego function (Fe) and its inverse (Fe-1)
- An optional stego-key (K) or password may be used to hide and unhide the message .

Cryptography algorithms in the image archive have been widely developed. Meanwhile, cryptography algorithms in audio archive are relatively few. In recent years there are so many algorithm have been developed to provide more security, enhanced quality with easy implementation and faster calculations. Among them all of the techniques have their own drawbacks like computational complexity, time consumption and reconstruction of secret information etc., Here, in this proposal we implemented pixel mapping based video steganography, which is a very simple and easy calculations and also provide more security.

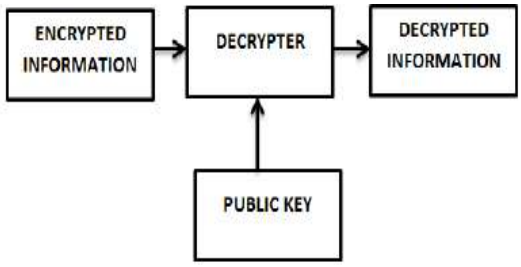


Fig.4.2.Cryptography decrypter

**Draw backs:**

1) Security is low.

2) Maximum capacity of data transfer does not possible.

3)In cryptography, in the encryption process, by using public key, any person can easily know the information.

4)In cryptography, we can hide the data and we can also know the data by applying reverse process.

# CHAPTER 5

# Proposed system

## 5.1 Proposed system

A survey of steganographic techniques reveals that there have been several techniques for hiding information or messages in host messages in such a manner that the embedded data should be imperceptible.

Substitution system substitutes redundant parts of a cover with a secret message. Spread spectrum techniques adopt ideas from spread spectrum communication. The statistical method encodes information by changing several statistical properties of a cover and use hypothesis testing in the extraction process. Distortion process stores information by signal distortion and measure the deviation from the original cover in the decoding step.

The cover generation method encodes information in the way a cover for secret communication is created. In case of hiding information in digital sound, phase Coding embeds data by altering the phase in a predefined manner. To a certain extent, modifications of the phase of a signal cannot be perceived by the human auditory system (HAS).

All these steganographic techniques deal with a few common types of steganography procedure depending on the variation of the host media. That means the cover object or the carrier object which will be used to hide the secret data. Different media like image, text, video and audio has been used as a carrier or host media in different times.

Using audio file as a cover object directs to Audio steganography. Practical audio embedding systems face hard challenges in fulfilling all three requirements due to the large power and dynamic range of hearing, and the large range of audible frequency of the .

The human auditory system (HAS) perceives sounds over a range of power greater than 109:1 and a range of frequencies greater than 103:1. The sensitivity of the HAS to the Additive White Gaussian Noise (AWGN) is high as well; this noise in a sound file can be detected as low as 70 dB below ambient level. On the other hand, opposite to its large dynamic range, HAS contains a fairly small differential range, i.e. loud sounds generally tend to mask out weaker sounds.

Additionally, HAS is insensitive to a constant relative phase shift in a stationary audio signal and some spectral distortions interprets as natural, perceptually non-annoying ones. Two

properties of the HAS dominantly used in steganographic techniques are frequency masking and temporal masking.

The concept using the perceptual holes of the HAS is taken from wideband audio coding (e.g. MPEG compression 1, layer 3, usually called mp3). In the compression algorithms, the holes are used in order to decrease the amount of the bits needed to encode audio signal, without causing a perceptual distortion to the coded audio. On the other hand, in the information hiding scenarios, masking properties are used to embed additional bits into an existing bit stream, again without generating audible noise in the audio sequence used for data hiding.

Some of the audio steganographic techniques are Lossless Adaptive Digital Audio Steganography, LSB based Audio Steganography, Audio Steganography using bit modification etc.

Some of the considerations to be kept in mind for efficient steganography techniques

- The cover media should not be degraded by the embedded data.
- It should appear that the cover media does not look distorted. It should not have a noticeable change in color composition, or that of the luminance. Frequently a cover media's size becomes enlarged; this can look very suspicious.
- Embedded data should be directly embedded into the cover media not in the header or wrapper. The embedded data needs to remain intact across different file formats.
- The embedded data should be immune to any manipulation. This ranges from any intentional modification or any modification through transmission.
- Error correcting codes should be inserted in the cover media to ensure integrity of data when or if the cover media is modified or tampered with.
- The embedded data should be recoverable and intact if only fragments of the cover media remain.

## 5.2 LSB based Audio Steganography:

In the current Endeavour, an audio file with ".wav" extension has been selected as host file. It is assumed that the least significant bits of that file should be modified without degrading the sound quality. To do that, first one needs to know the file structure of the audio file. Like

most files, WAV files have two basic parts, the header and the data. In normal wav files, the header is situated in the first 44 bytes of the file. Except the first 44 bytes, the rest of the bytes of the file are all about the data. The data is just one giant chunk of samples that represents the whole audio. While embedding data, one can't deal with the header section. That is because a minimal change in the header section leads to a corrupted audio file.

A program has been developed which can read the audio file bit by bit and stores them in a different file. The first 44 bytes should be left without any change in them because these are the data of the header section.

Then start with the remaining data field to modify them to embed textual information. For example, if the word "Audio" has to be embedded into an audio file one has to embed the binary values of the word "Audio" into the audio data field.

Consider the following table:

**TABLE 5.1**

**LETERS WITH ASCII VALUES AND CORRESPONDING BINARY VALUES :**

| Letter | ASCII Value | Corresponding Binary Value |
|---|---|---|
| A | 065 | 01000001 |
| U | 117 | 01110101 |
| D | 100 | 01100100 |
| I | 105 | 01101001 |
| O | 111 | 01101111 |

From the table, one can come to a point that to embed the word "Audio" into the host audio file actually the corresponding eight-bit binary values have to be embedded into the data field of that audio file.

## 5.3 BLOCK DIAGRAMS:

### EMBEDDING PROCESS:

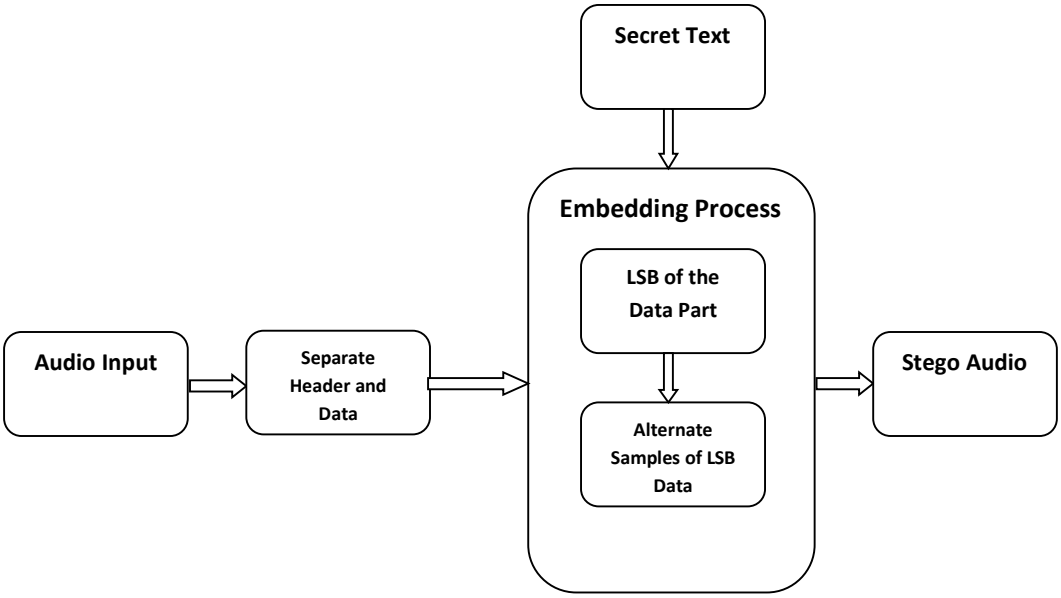


**FIG.5.1 : Block diagram of embedding process**

### EXTRACTION PROCESS:

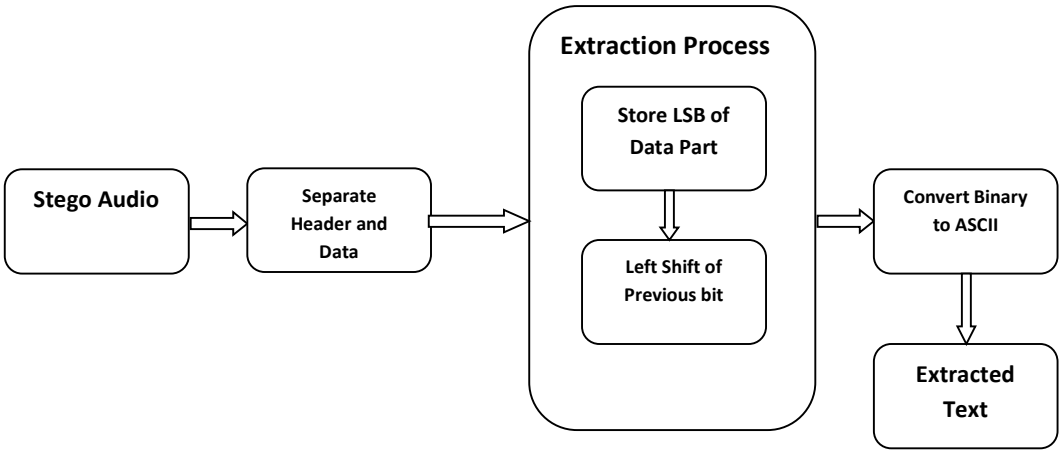


**FIG 5.2: Block diagram of Extraction Process**

## 5.3.1 EMBEDDING PROCESS DESCRIPTION:

By considering the Block diagram of embedding process, an audio file named "audio.wav" has been selected for this experiment in the first block of the embedding process. After checking the binary values of each sample, the samples were separated as first 44 samples were left without any changes for the header of the audio and remaining samples were left for the data part of the audio file. An audio file named "audio.wav" has been selected for this experiment. After checking the binary values of each sample, first 44 samples were left without any changes.

The data embedding with LSB modification has been started after the header section. If the data embedding process is started from 51st sample then the LSB value of the 51st sample should be modified. If the binary value of the corresponding sample is "01110100" then "1" should be modified. From Table I it can be observed that to embed the letter "A", the sender has to embed the binary value "01000001". That is why according to the embedding algorithm "A" should be embedded according to Table 5.1.

**TABLE 5.1**

**SAMPLES OF AUDIO FILE WITH BINARY VALUES BEFORE AND AFTER EMBEDDING**

| Sample No. | Binary values to corresponding samples | Binary value to be embedded | Binary values after modification |
|---|---|---|---|
| 51 | 01110100 | 0 | 01110100 |
| 53 | 01011110 | 1 | 01011111 |
| 55 | 10001011 | 0 | 10001010 |
| 57 | 01111011 | 0 | 01111010 |
| 59 | 10100010 | 0 | 10100010 |
| 61 | 00110010 | 0 | 00110010 |
| 63 | 11101110 | 0 | 11101110 |
| 65 | 01011100 | 1 | 01011101 |

According to the same way the remaining consecutive letters of the word “Audio” is embedded in the file “audio.wav.”

Editing of the existing binary values with the intended binary values causes a minimal change in the audio file “audio.wav” that remains almost imperceptible to anyone other than the sender. When it comes to the point of data retrieving at the receiver’s end, the retrieving algorithm has to be followed:

**Algorithm (For Embedding of Data):**

• Leave the header section of the audio file untouched...

• Start from a suitable position of the data bytes. (For the experiment purpose the present start byte was the 51st byte). Edit the least significant bit with the data that have to be embedded.

• Take every alternate sample and change the least significant bit to embed the whole message. The data retrieving algorithm at the receiver’s end follows the same logic as the embedding algorithm.

First, change the audio message into binary format that has come from the source as stego-object. Leave first 50 bytes with no change in them. Start from 51st bit, check the least significant bit, and store it in a queue. Check every alternate sample to collect the whole messages. Like 53rd, 55th and 57th and so on. Store the least significant bits of the alternate samples in the queue with left shift of previous bit. Convert the binary values to decimal to get back the ASCII from which the text can be retrieved. The whole retrieval process can be depicted with the following table more thoroughly:

The data embedding with LSB modification has been started after the header section. If the data embedding process is started from 51st sample then the LSB value of the 51st sample should be modified. If the binary value of the corresponding sample is “01110100” then “1” should be modified. It can be observed that to embed the letter “A”, the sender has to embed the binary value “01000001”. That is why according to the embedding algorithm “A” should be embedded.

According to the same way the remaining consecutive letters of the word “Audio” is embedded in the file “audio.wav.”

Editing of the existing binary values with the intended binary values causes a minimal change in the audio file “audio.wav” that remains almost imperceptible to anyone other than the sender.

The output of the embedding process after embedding the audio file of .wav extension with respect to the AUDIO text file is named as stego audio.

Thus the basic steps taken here are firstly selecting the audio file of .wav format and separating the header and data of the audio with corresponding samples. Then by considering the text which has to be embedded has to be embedded corresponding to the LSB of the audio samples by modifying the bits. The embedded audio file is finally represented as a Stego Audio.

### 5.3.2 EXTRACTION PROCESS DESCRIPTION :

When it comes to the point of data retrieving at the receiver’s end, the retrieving algorithm has to be followed:

First, change the audio message into binary format that has come from the source as stego-object. Leave first 50 bytes with no change in them. Start from 51st bit, check the least significant bit, and store it in a queue. Check every alternate sample to collect the whole messages. Like 53rd, 55th and 57th and so on. Store the least significant bits of the alternate samples in the queue with left shift of previous bit. Convert the binary values to decimal to get back the ASCII from which the text can be retrieved. The whole retrieval process can be depicted with the following table more thoroughly:

In the embedding process of the letter “A” was stated that is why, the retrieval process of “A” is depicted. Starting from the 51st sample, every alternate sample has been checked and the least significant bit has been stored into a queue with a left shift of previous bit.

After getting all the bits in the queue, start from the left hand side, take 8 bits and convert them into equivalent decimal to get the ASCII, from the ASCII retrieve the embedded textual message.

## 5.4 EXRACTION OF DATA FROM AUDIO FILE:

**Algorithm (For Extracting of Data):**

•Leave first 50 bytes.

• Start from the 51st byte and store the least significant bit in a queue.

• Check every alternate sample and store the least significant bit in the previous queue with a left shift of the previous bit.

• Convert the binary values to decimal to get the ASCII values of the secret message.

• From the ASCII find the secret message.

**TABLE 5.2**

| Sample No. | Binary values with embedded secret data | Bits that are stored in the Queue |
|---|---|---|
| 51 | 01110100 | 0 |
| 53 | 01011111 | 01 |
| 55 | 10001010 | 010 |
| 57 | 01111010 | 0100 |
| 59 | 10100010 | 01000 |
| 61 | 00110010 | 010000 |
| 63 | 11101110 | 0100000 |
| 65 | 01011101 | 01000001 |

From the extraction process, it is clearly observed that after getting 01000001 in the queue it is converted into the equivalent decimal that is 65, the ASCII of “A”. Thus “A” is retrieved. Like the same way, the next letters also have been retrieved and hence the complete word “Audio.”

Thus the basic steps taken here is to considering the stego audio and separating the header and data samples and storing the LSB of the stego audio data part in a queue and left shifting the corresponding previous bits of the sample. Then the final data from the queue is converted in to the ASCII with respect to the binary values of the sample in the queue and the text is extracted.

To develop this algorithm multiple bits of each sample of the file have been changed or modified to insert text data in it. It has also been observed the degradation of the host audio file after modification of the bits. The bit modification was done by various ways, like 1, 2, 3, 4 bits were changed in turn. But after going through all the modification it has been observed that 1 bit change in LSB gave the best result.

Thus, data can be embedded according to the following algorithm.

# CHAPTER 6

# MATLAB

## 6.1 INTRODUCTION

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

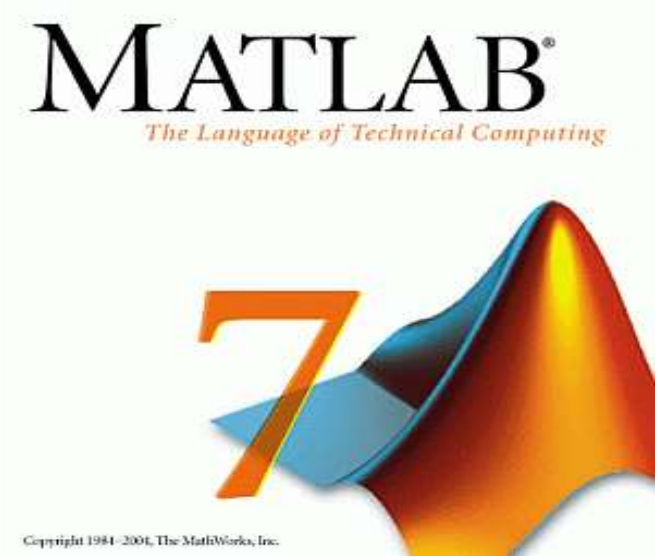


Figure 6.1: MATLAB

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

## 6.2 THE MATLAB SYSTEM

The MATLAB system consists of five main parts

**DEVELOPMENT ENVIRONMENT**

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

**THE MATLAB MATHEMATICAL FUNCTION LIBRARY**

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

**THE MATLAB LANGUAGE :**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

In the item part MATLAB programming and the highlight, which is to be settled, is the base need. A rate of the points of interest from MATLAB in highlight taking care of are:

- Easy to work with; as Images are lattices
- Built in capacities for complex operations and calculations (Ex. FFT, DCT, and so forth)
- Image transforming tool kit,
- Supports most picture configurations (.bmp, .jpg, .gif, tiff and so forth)

## 6.3 INTRODUCTION TO MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB stands for matrix laboratory, and was written originally to provide easy access to matrix software developed by LINPACK (linear system package) and EISPACK (Eigen system package) projects. MATLAB is therefore built on a foundation of sophisticated matrix software in which the basic element is array that does not require pre dimensioning which to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of time.

MATLAB features a family of applications specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow learning and applying specialized technology. These are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control system, neural networks, fuzzy logic, wavelets, simulation and many others.

Typical uses of MATLAB include: Math and computation, Algorithm development, Data acquisition, Modeling, simulation, prototyping, Data analysis, exploration, visualization,

Scientific and engineering graphics, Application development, including graphical user interface building.

MATLAB is a program that was originally designed to simplify the implementation of numerical linear algebra routines. It has since grown into something much bigger, and it is used to implement numerical algorithms for a wide range of applications. The basic language used is very similar to standard linear algebra notation, but there are a few extensions that will likely cause you some problems at first.

## 6.3.1 BASIC BUILDING BLOCKS OF MATLAB

The basic building block of MATLAB is MATRIX. The fundamental data type is the array. Vectors, scalars, real matrices and complex matrix are handled as specific class of this basic data type. The built in functions are optimized for vector operations. No dimension statements are required for vectors or arrays.

**MATLAB WINDOW**

The MATLAB works based on five windows: Command window, Workspace window, Current directory window, Command history window, Editor Window, Graphics window and Online-help window.

**COMMAND WINDOW**

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. It is opened when the application program is launched. All commands including user-written programs are typed in this window at MATLAB prompt for execution.

**WORK SPACE WINDOW**

MATLAB defines the workspace as the set of variables that the user creates in a work session. The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information.

### CURRENT DIRECTORY WINDOW

The current Directory tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory "work" is a subdirectory of the main directory "MATLAB"; which is installed in drive C. Clicking on the arrow in the current directory window shows a list of recently used paths. MATLAB uses a search path to find M-files and other MATLAB related files. Any file run in MATLAB must reside in the current directory or in a directory that is on search path.

### COMMAND HISTORY WINDOW

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands. This is useful to select various options in addition to executing the commands and is useful feature when experimenting with various commands in a work session.

### EDITOR WINDOW

The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The editor can appear in a window by itself, or it can be a sub window in the desktop. In this window one can write, edit, create and save programs in files called M-files.

MATLAB editor window has numerous pull-down menus for tasks such as saving, viewing, and debugging files. Because it performs some simple checks and also uses color to differentiate between various elements of code, this text editor is recommended as the tool of choice for writing and editing M-functions.

## GRAPHICS OR FIGURE WINDOW

The output of all graphic commands typed in the command window is seen in this window.

## ONLINE HELP WINDOW

MATLAB provides online help for all it's built in functions and programming language constructs. The principal way to get help online is to use the MATLAB help browser, opened as a separate window either by clicking on the question mark symbol (?) on the desktop toolbar,

or by typing help browser at the prompt in the command window. The help Browser is a web browser integrated into the MATLAB desktop that displays a Hypertext Markup Language (HTML) documents. The Help Browser consists of two panes, the help navigator pane, used to find information, and the display pane, used to view the information. Self-explanatory tabs other than navigator pane are used to perform a search.

## 6.3.2 MATLAB FILES

MATLAB has three types of files for storing information. They are: M-files and MAT-files.

**1. M-Files**

These are standard ASCII text file with 'm' extension to the file name and creating own matrices using M-files, which are text files containing MATLAB code. MATLAB editor or another text editor is used to create a file containing the same statements which are typed at the MATLAB command line and save the file under a name that ends in .m. There are two types of M-files:

**2. SCRIPT FILES**

It is an M-file with a set of MATLAB commands in it and is executed by typing name of file on the command line. These files work on global variables currently present in that environment.

**3. FUNCTION FILES**

A function file is also an M-file except that the variables in a function file are all local. This type of files begins with a function definition line.

## MAT-FILES

These are binary data files with .mat extension to the file that are created by MATLAB when the data is saved. The data written in a special format that only MATLAB can read. These are located into MATLAB with 'load' command.

## THE MATLAB SYSTEM

The MATLAB system consists of five main parts:

**DEVELOPMENT ENVIRONMENT**

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

**THE MATLAB MATHEMATICAL FUNCTION**

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

**MATLAB LANGUAGE**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

**GRAPHICS**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

**THE MATLAB APPLICATION PROGRAM INTERFACE (API)**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

## 6.4 GRAPHICAL USER INTERFACE (GUI)

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type guide filename where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.

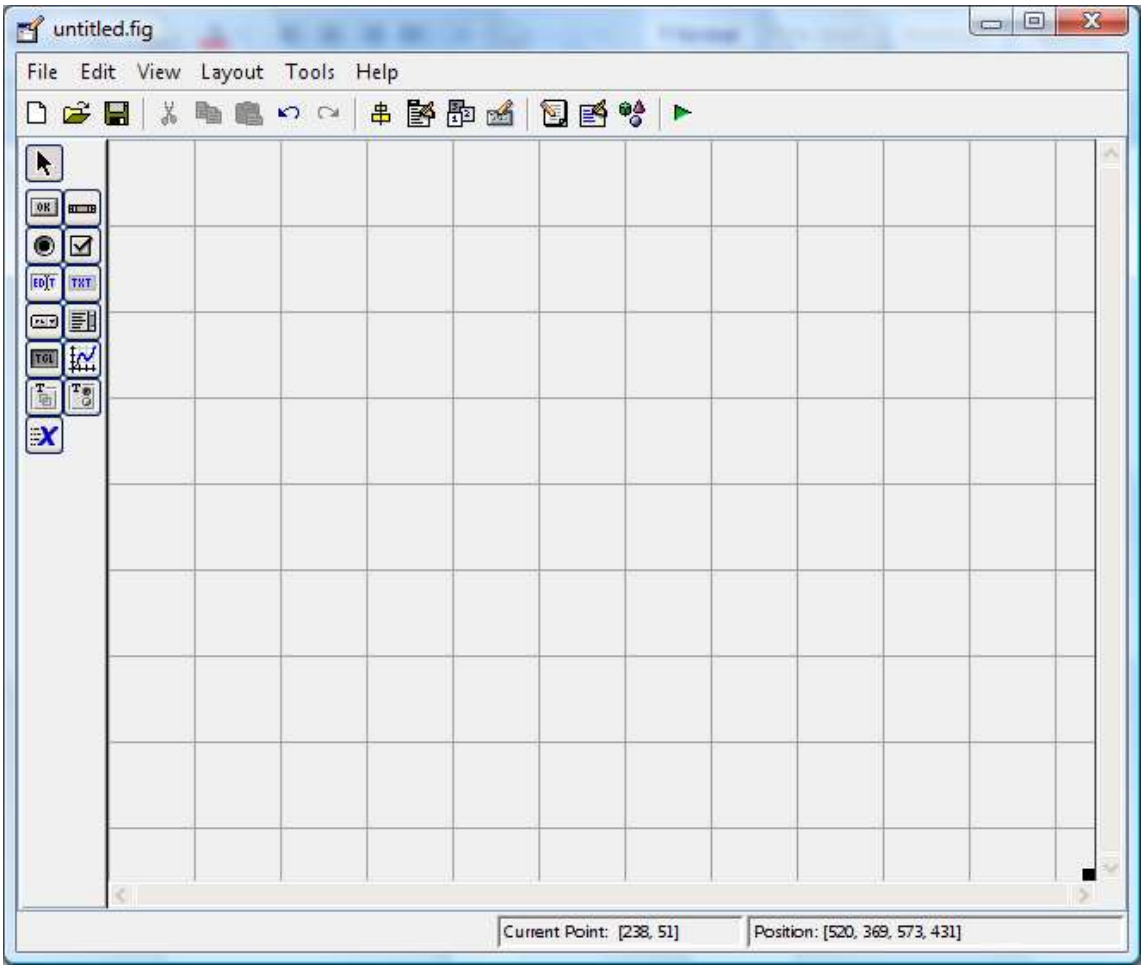


Figure 6.2: GUIDE new blank window

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

# Results

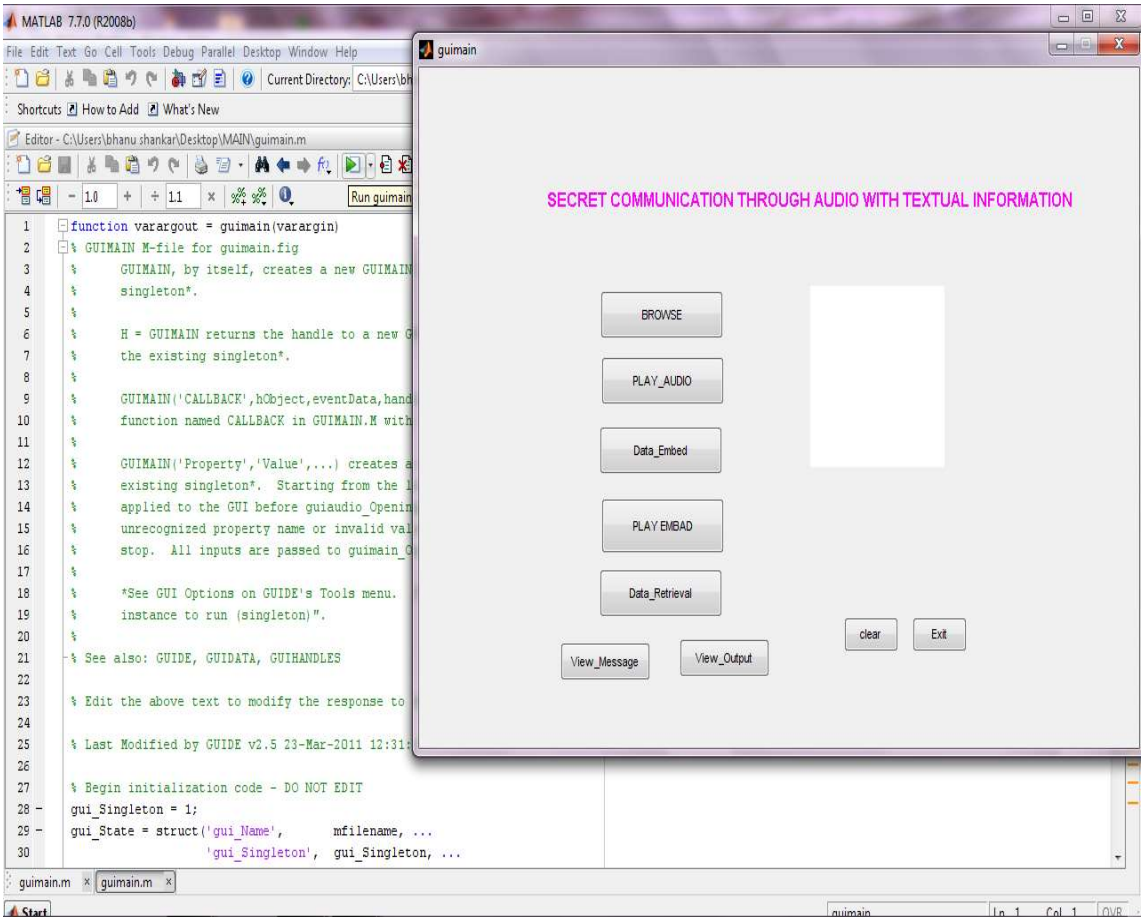


Figure 6.3: creating GUI panel

By running the source code of the project using run button from the Matlab we get the above GUI panel having various task buttons like browse, play audio, data embed, play embed audio, data retrieval etc.

## Creating Message file and Viewing the file:

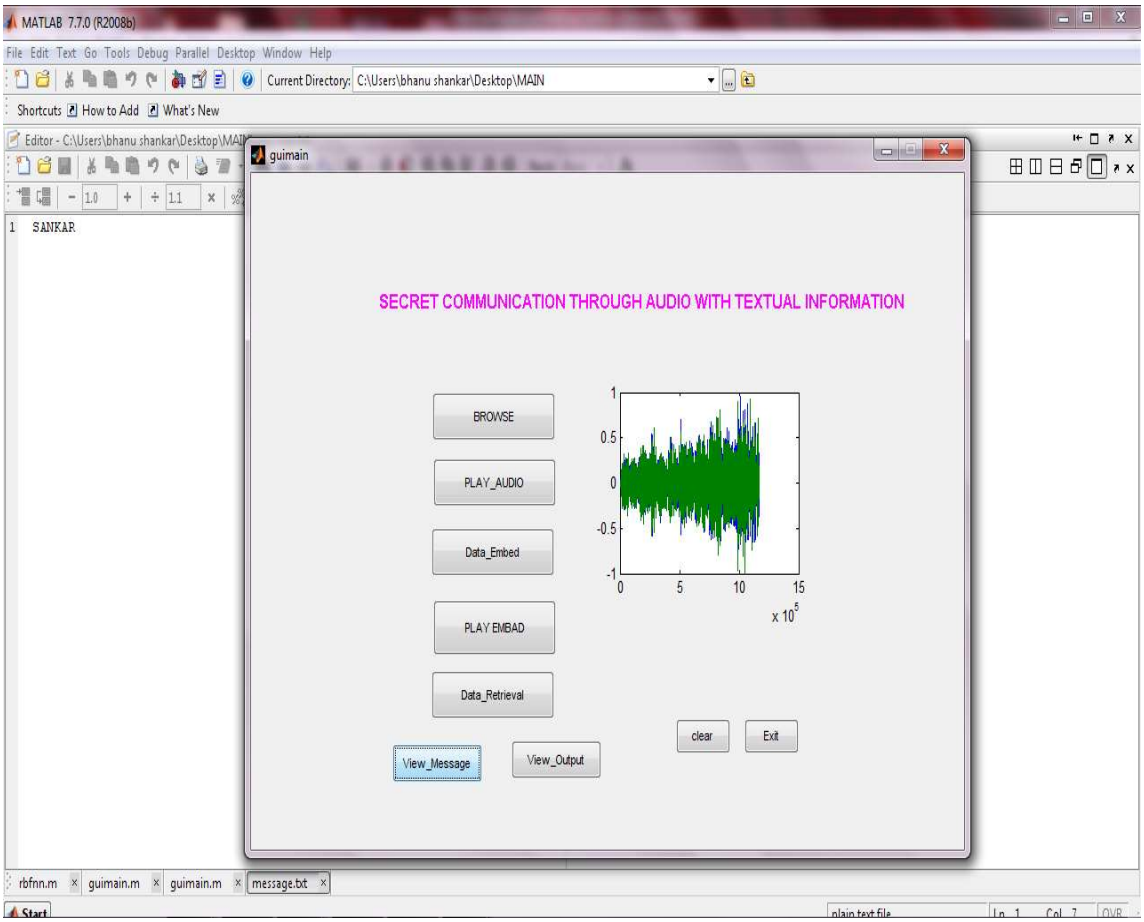


Figure 6.4 : Creating Message file and Viewing the file

To view the text message from the GUI panel we can select the View Message button.

## Selecting the Audio file:

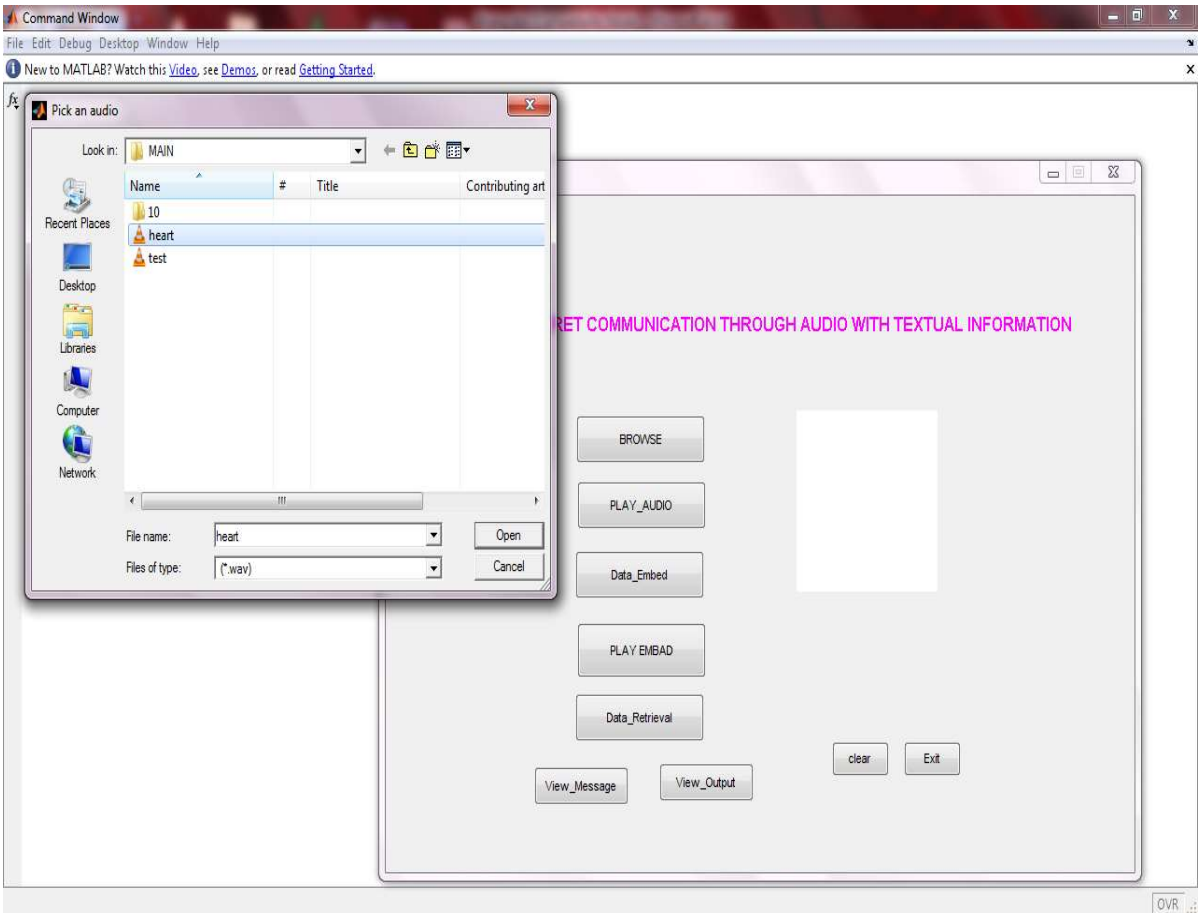


Figure 6.5: Selecting the Audio file

To select the Audio file we have to select the Browse from GUI and select the audio file with .wav extension in the respected Directory.

## Playing and Plotting the Audio file:

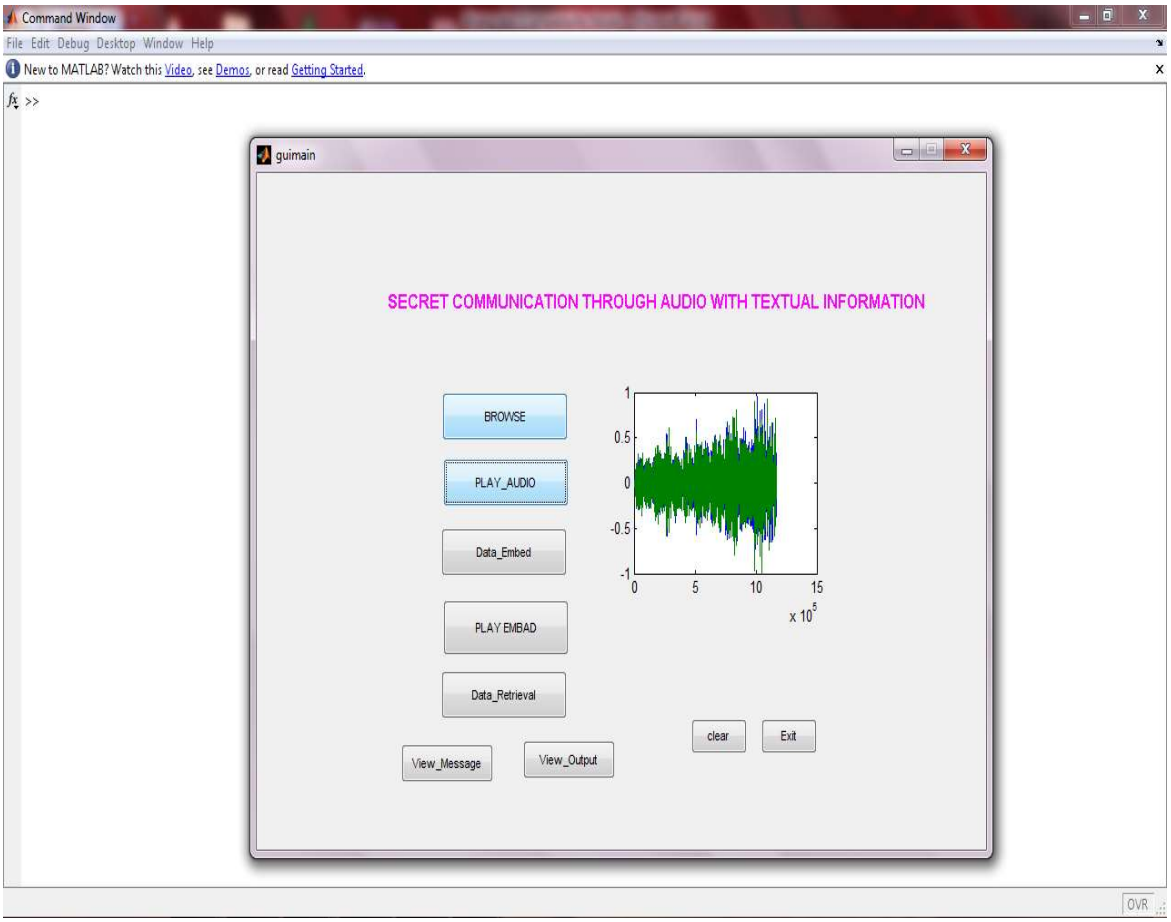


Figure 6.6: Playing and Plotting the Audio file

By selecting the Play Audio Button we can play the selected audio file and after playing the audio we get the corresponding plot of the audio.

## Embedding the Data:

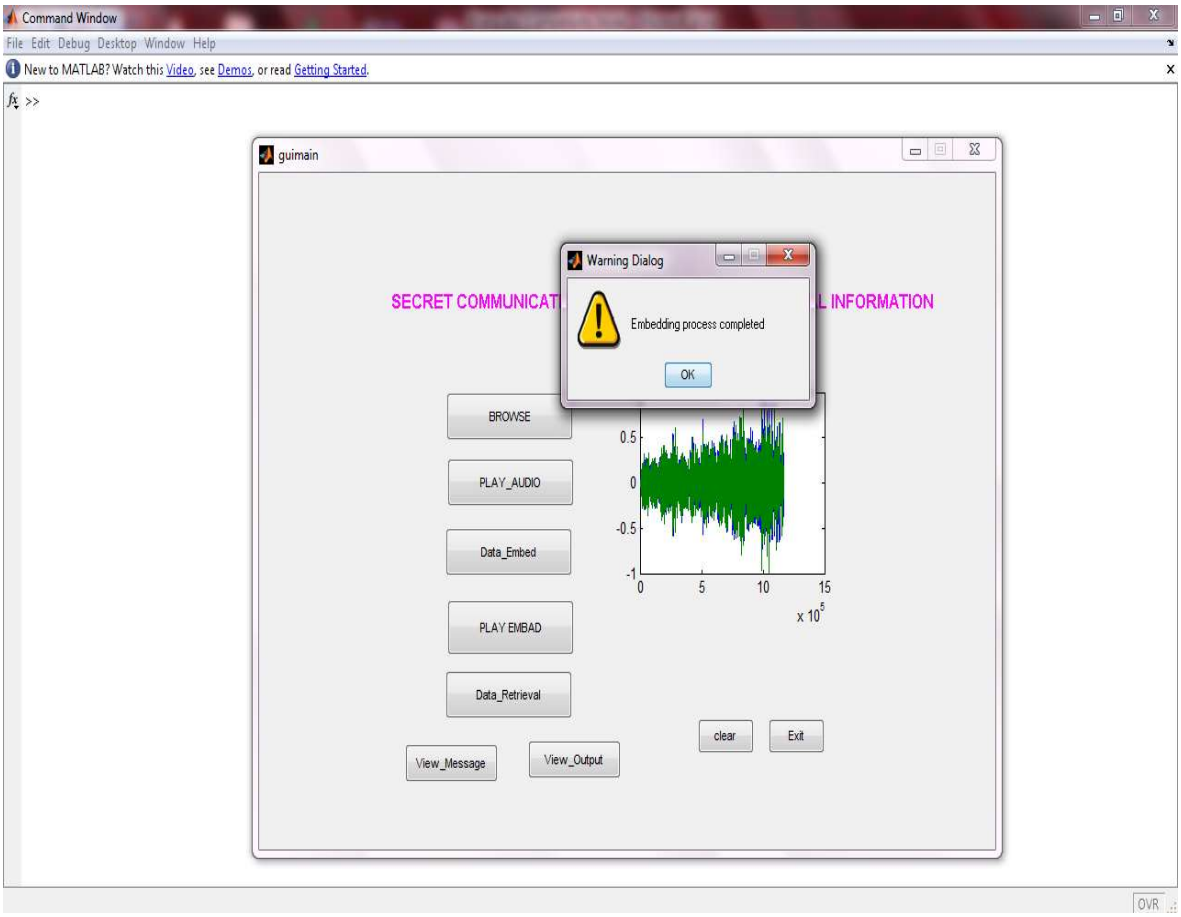


Figure 6.7: Embedding the Data

By selecting the Data embed the data is embedded with respect to the audio file and a warning dialog is displayed as embedding process is completed.

## Playing the Embedded Audio File:

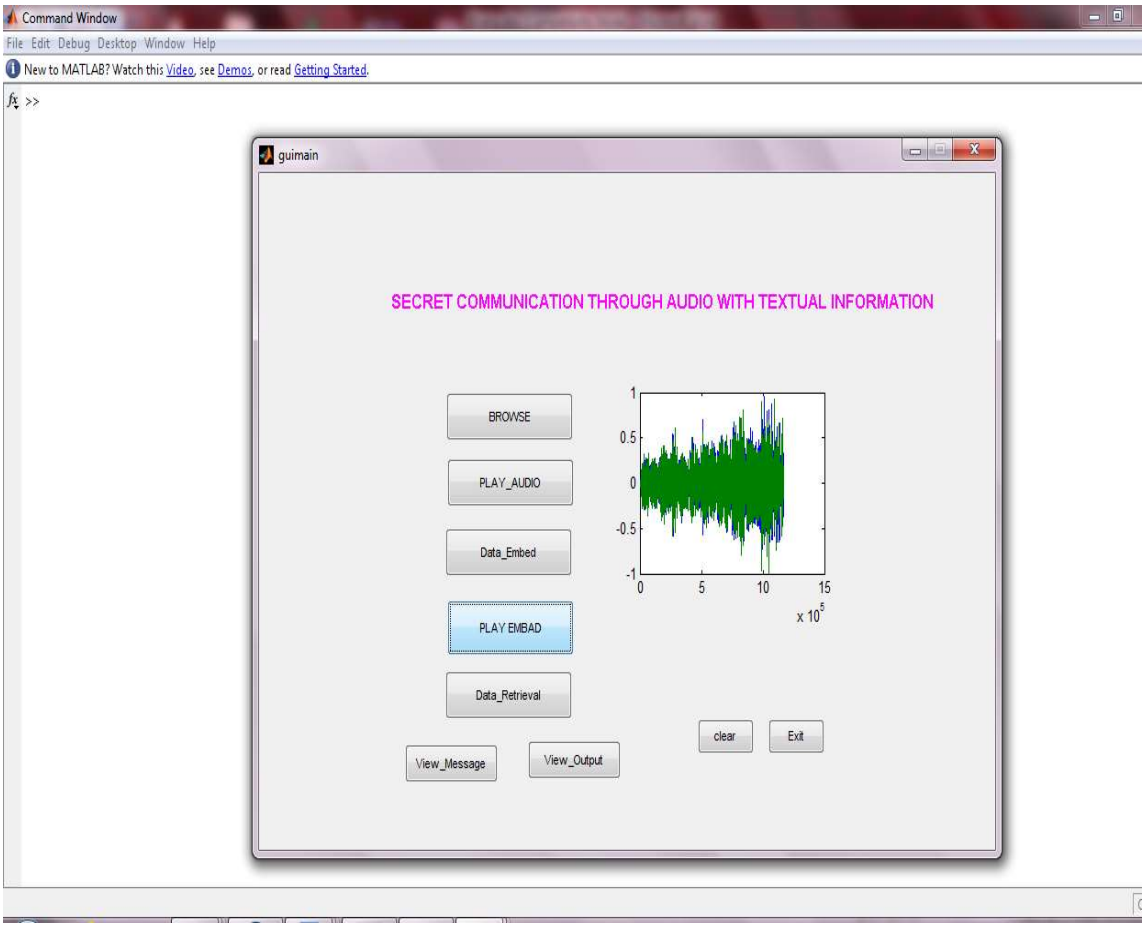


Figure 6.8: Playing the Embedded Audio File

The embedded Audio file is played after selecting the Play embed and the graph of the embedded audio file is plotted.

## Retrieving the Data:

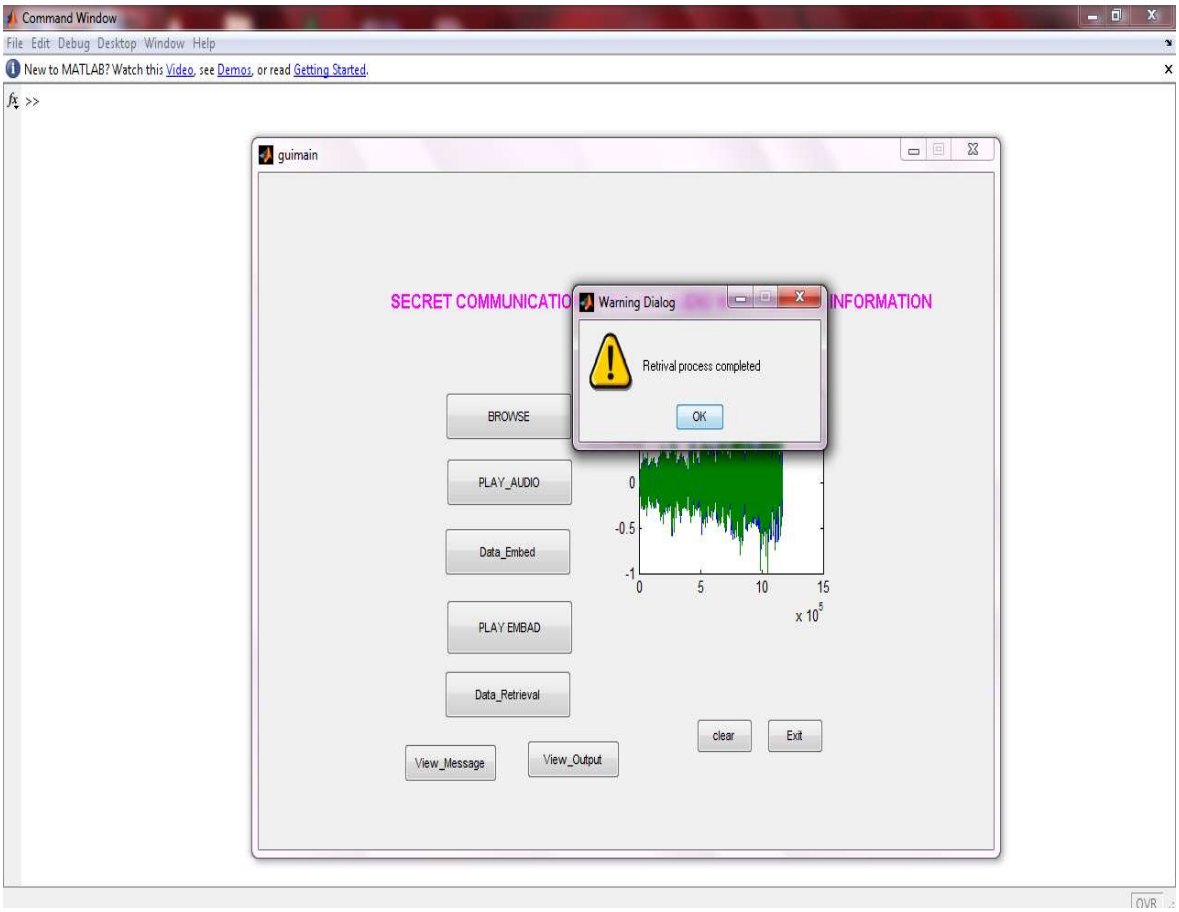


Figure 6.9: Retrieving the Data

The final data can be retrieved from the embedded audio file with respect to the Data retrieve Button and the output file can be viewed from the view output as shown below.

## Viewing the Retrieval Data:

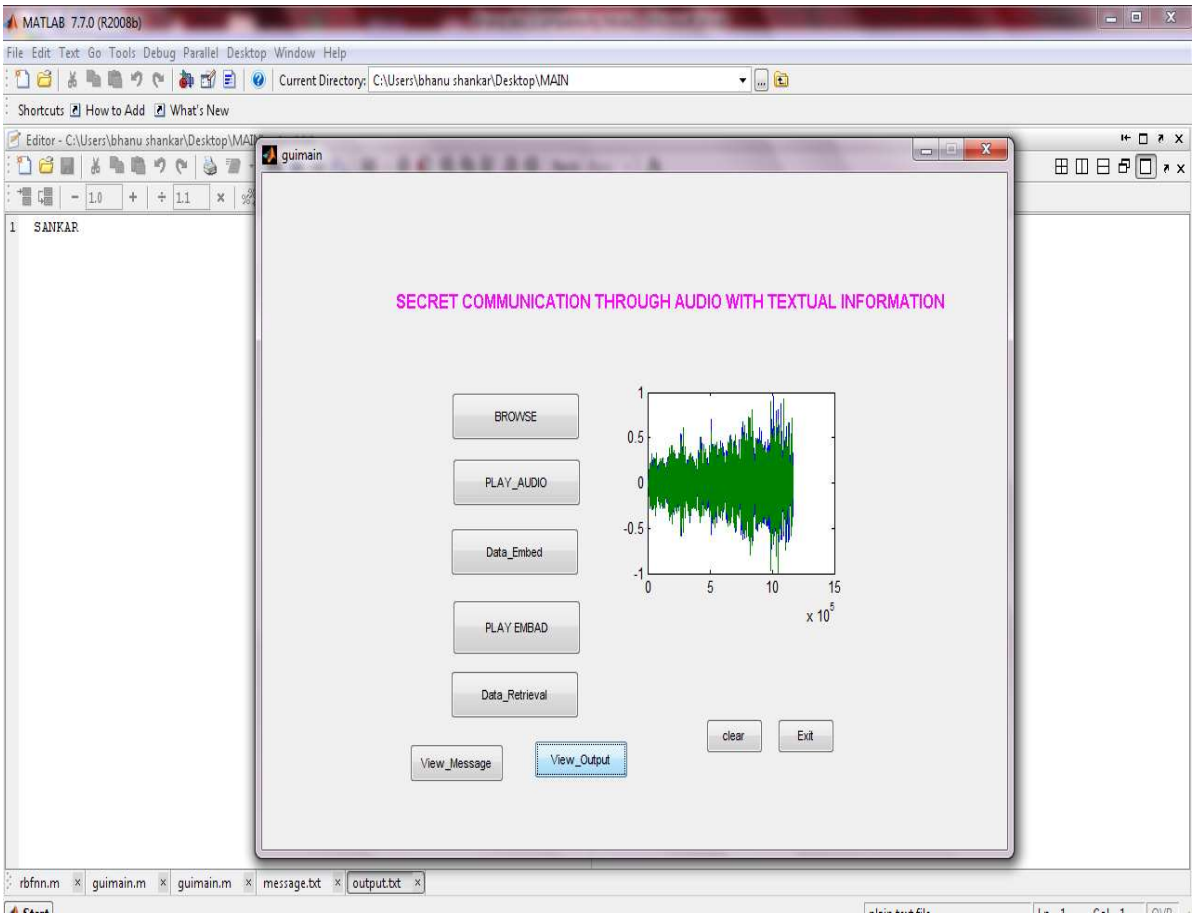


Figure 6.10: Viewing the Retrieval Data

Final retrieved message can be view by using view output button.

## APPLICATIONS

**Usage in modern printers**:

**Defense Applications**: Steganography is mostly used in Defense Applications.

**Alleged use by intelligence services:**

# CONCLUSION:

A method of embedding text-based data into a host audio file using the method of bit modification has been presented in this paper. A procedure has been developed in which the data field is edited to embed intended data into the audio file. To proceed with this, the header section of the audio has been checked perfectly because a minimal change in the header section may leads to a corruption of whole audio file.

In this algorithm, as an experiment first 50 bytes have been left untouched and starting from the 51st bytes every alternate sample has been modified to embed textual information. How the performance is affected by changing different bit fields has not been reported in this work. However a rough study was made to see how the changing of a specific bit field creates degradation in the host audio file and in which point it leads to perceptible change in the audible sound quality to any other third party other than the sender or receiver. It was noticed that changing the least significant bit of the bytes gave the best results.

# REFERENCES

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding", IBM Systems Journal, vol. 35, Issues 3&4, 1996, pp. 313-336.

[2] Kharrazi, M., Sencar, Husrev T., and Memon, N., “Image Steganography: Concepts and Practice”, WSPC, April 22, 2004.

[3] Stefan Katzenbeisser, Fabien A. P. Petitcolas, “Information Hiding Techniques for Steganography and Digital Watermarking”. Boston, Artech House, pp. 43 – 82. 2000.

[4] K. Matsui and K. Tanaka. Video-steganography. In: IMA Intellectual Property Project Proceedings, volume 1, pp 187-206, 1994.

[5] N.F. Johnson and S. Jajodia, “Exploring Steganography: Seeing the Unseen,” Computer, vol. 31, no. 2, pp. 26-34, IEEE, Feb. 1998.

[6] Matsuoka, H., “Spread Spectrum Audio Steganography using Sub – band Phase Shifting”, Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP' 06), IEEE, 2006.

[7] S.S. Agaian, D. Akopian, O. Caglayan, S. A. D’Souza, “Lossless Adaptive Digital Audio Steganography,” In Proc. IEEE Int. Conf. Signals, Systems and Computers, pp. 903-906, November 2005.

[8] K. Gopalan, “Audio steganography using bit modification”, Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Vol. 2, pp. 421424, April 2003.

[9] Mohammad Pooyan, Ahmed Delforouzi, “LSB-based Audio Steganography Method Based on Lifting Wavelet Transform”, International Symposium on Signal Processing and Information Technology, IEEE, 2007.

[10] C. Chang, T. S. Chen and H. S. Hsia, “An Effective Image Steganographic Scheme Based on Wavelet Transform and Pattern- Based Modification”, IEEE Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing, 2003.

[11] Chen and G.W. Womell, “Quantization index modulation: a class of provably good methods for digital watermarking and information embedding”, IEEE Transactions on Information Theory, Vol. 47, No. 4, pp. 1423-1443, May 2001.

[12] B. Chen, “Design and analysis of digital watermarking, information embedding, and data hiding systems,” Ph.D. dissertation, MIT, Cambridge, MA, June 2000.

[13] M. M. Amin, M. Salleh, S. Ibrahim, M. R. Katmin and M. Z. I. Shamsuddin, “Information Hiding using Steganography”,4th National Conference on Telecommunication Technology Proceedings, Shah Alam, Malaysia, IEEE, 2003.

[14] J. Zollner, H. Federrath, H. Klimant, et al., “Modelling the Security of Steganographic Systems”, in 2nd Workshop on Information Hiding, Portland, April 1998, pp. 345-355.

[15] Johnson, Neil F. and Stefan Katzenbeisser. “A Survey of Steganographic Techniques”, In Information Hiding: Techniques for Steganography and Digital Watermarking. Boston, Artech House. 43-78. 2000.

[16] Y. Yardimci, A. E. Cetin and R. Ansari, “Data hiding in speech using phase coding”, ESCA. Eurospeech97, pp. 1679-1682, Greece, Sept. 1997.

[17] K. Bennett, “Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text”, Purdue University, CERIAS Tech. Report 2004-13, 2004.

[18] Matsuoka, H., “Spread Spectrum Audio Steganography using Sub – band Phase Shifting”, Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'06), IEEE, 2006.

[19] Noll P., “Wideband speech and audio coding”. IEEE Communications Magazine 31(11): 1993, pp 34–44.

[20] “Introduction to Digital Speech Processing” Lawrence R. Rabiner[1] and Ronald W. Schafer[2]

[21] Basic MATLAB, Simulink and Stateflow by **Richard Colgren**, University of Kansas ( *AIAA Education Series* ’07), First Edition, 485 pages.

[22] Bamford, J. (2001). *Body of Secrets : Anatomy of the Ultra-Secret National Security Agency from the Cold War Through the Dawn of a New Century.* New York: Doubleday.

[23] Barr, T.H. (2002). *Invitation to Cryptology.* Upper Saddle River, NJ: Prentice Hall.

[24] Bauer, F.L. (2002). *Decrypted Secrets: Methods and Maxims of Cryptology,* 2nd ed. New York: Springer Verlag.