Abstract

Data is an important asset for any individual or organization and must be protected from intruders or hackers. The need to hide data from hackers has existed since ancient times, and nowadays, there are developments in digital media, such as audio, video, images, and so on. To secure secret information, different media methods are used and steganography is one. Steganography hides the data under other data without any differentiable changes. Many individual steganography tools can be used to transfer data securely and, in this paper, a new tool is proposed that decreases time and effort. Using this tool, we hide the text in audio, video, or images in one place, so there was no need to have access to multiple tools. This proposed tool developed using the least significant bit (LSB) approach.
# TABLE OF CONTENTS

Abstract.......................................................................................................................... ii

Table of Contents............................................................................................................. iii

List of Figures..................................................................................................................... v

1. Introduction.....................................................................................................................1
   1.1 Hiding Data .............................................................................................................. 1
       1.1.1 Cryptography vs. Steganography ................................................................. 1
   1.2 Steganography ....................................................................................................... 1
   1.3 Importance of Steganography ............................................................................... 4
   1.4 Reasons for Steganography .................................................................................. 5
   1.5 Algorithms Used for Hiding Data ......................................................................... 5
   1.6 Objective ............................................................................................................... 6

2. Previous Research, Motivation, Scope, and Proposed Research ......................... 7
   2.1 Previous Research ................................................................................................. 7
   2.2 Motivation ............................................................................................................. 8
   2.3 Scope .................................................................................................................... 8
   2.4 Proposed Research ............................................................................................... 8

3. Architecture ............................................................................................................... 9

4. Design Approach ........................................................................................................ 10
   4.1 Hiding the secret file ............................................................................................ 11
   4.2 Extracting the secret file ...................................................................................... 12
   4.3 Software Requirements ...................................................................................... 13
   4.4 UML Diagrams .................................................................................................... 13
4.4.1 Class Diagram .................................................................13
4.4.2 Use Case Diagram ............................................................14
4.4.3 Sequence Diagram ............................................................15
  4.4.3.1 Images ........................................................................15
  4.4.3.1 Audio .........................................................................15
  4.4.3.1 Video .........................................................................16
4.4.4 Component Diagram ..........................................................17
4.4.5 Collaboration Diagram .......................................................18
  4.4.5.1 Images ........................................................................18
  4.4.5.2 Audio .........................................................................18
  4.4.5.3 Video .........................................................................19
4.4.6 Deployment Diagram ..........................................................19
5. Implementation ........................................................................21
  5.1 Hiding the Secret Data in Digital Files ...............................21
    5.1.1 Hiding Text in Images ..................................................25
    5.1.2 Hiding Text in Audio ....................................................26
    5.1.3 Hiding Text in Video ....................................................27
  5.2 Extracting the Secret Data from Digital Files ...................28
    5.2.1 Extracting Text from Images .......................................30
    5.2.2 Extracting Text from Audio ..........................................31
    5.2.3 Extracting Text from Video ..........................................32
6. Testing and Results ...............................................................34
7. Conclusion .........................................................................................................................43
8. Future Work.......................................................................................................................44

Bibliography and References .................................................................................................45
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>An Example of Steganography</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Pure Steganography</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Example of LSB</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Architecture of Proposed Steganography</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>Hiding Secret Messages in Digital Files</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>Extracting Secret Message from Digital Files</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>Class Diagram</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Use Case Diagram</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>Sequence Diagram for Image</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>Sequence Diagram for Audio</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>Sequence Diagram for Video</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>Component Diagram</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>Collaboration Diagram for Image</td>
<td>18</td>
</tr>
<tr>
<td>14</td>
<td>Collaboration Diagram for Audio</td>
<td>18</td>
</tr>
<tr>
<td>15</td>
<td>Collaboration Diagram for Video</td>
<td>19</td>
</tr>
<tr>
<td>16</td>
<td>Deployment Diagram</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>Home Screen for Hiding Secret Data in Digital Files</td>
<td>21</td>
</tr>
<tr>
<td>18</td>
<td>Key File</td>
<td>22</td>
</tr>
<tr>
<td>19</td>
<td>Adding Digital File</td>
<td>22</td>
</tr>
<tr>
<td>20</td>
<td>Browsing and Naming a File</td>
<td>23</td>
</tr>
<tr>
<td>21</td>
<td>Hiding Text Successfully</td>
<td>24</td>
</tr>
<tr>
<td>22</td>
<td>Flow Diagram for Embedding Text in Images</td>
<td>25</td>
</tr>
</tbody>
</table>
Figure 23. Flow Diagram for Embedding Text in Audio (Wav, Midi) ........................................26
Figure 24. Flow Diagram for Embedding Text in Video ..........................................................27
Figure 25. Home Screen for Extraction ..................................................................................28
Figure 26. Adding a Stego File ...............................................................................................29
Figure 27. Successful Extraction Process ................................................................................30
Figure 28. Retrieving Text from Image ..................................................................................31
Figure 29. Retrieving Text from Audio ..................................................................................32
Figure 30. Retrieving Text from Video ..................................................................................33
Figure 31. Checking Hide Button .........................................................................................34
Figure 32. Checking Extract Button .....................................................................................35
Figure 33. Checking Text Options .........................................................................................36
Figure 34. Checking the Browse Option ................................................................................36
Figure 35. Testing + Button ....................................................................................................37
Figure 36. Testing the – Button .............................................................................................38
Figure 37. Testing the Cancel Option .....................................................................................38
Figure 38. Inputs for Hiding Text in Images ..........................................................................39
Figure 39. Output Screen for Hiding Text in Images .............................................................40
Figure 40. Inputs for Extraction of a Secret Message from a Stego-Image ............................40
Figure 41. Output Screen for Extraction the Secret Message from Stego Images ..............41
Figure 42. Hashing the Secret Message ................................................................................42

LIST OF TABLES

Table 1. Software Requirements .........................................................................................14
1. Introduction

Globalization has led to the rapid growth of the internet through which consumers can send and receive large amounts of data (e.g., text, audio, video, and images). In modern communication systems, securing data is of utmost importance. Yet sending and receiving secret files over the internet is still insecure, and therefore hiding data in an effective way protects this secret information.

1.1 Hiding Data:

Data is important to any organization. They must be protected from the unauthorized access. Data should only visible to the sender and receiver of transmitted data, and they should be hidden from hackers. Hiding data is nothing more than protecting the data in some medium or encrypting the data. There are many techniques that use the concept of hiding data; cryptography and steganography are among them.

1.1.1 Cryptography vs. Steganography

Steganography and cryptography are closed related, with the main difference being their goals. Cryptography encrypts the data, which makes it unreadable, but the encrypted data cannot be hidden from unauthorized users as presence of hidden data is known. In contrast, steganography prohibits unauthorized users from even having any knowledge of the existence of the hidden data.

1.2 Steganography

Steganography is the art or practice of hiding a message, audio, video, image, or file within any of these formats. The word “steganography” is from Greek and means “covered writing.” In this paper, we focus on hiding text in multimedia files such as
audio, video, and images. Figure 1 is an example of steganography which clearly shows that some data is hidden in image.

![Figure 1. An Example of Steganography](image)

Steganography is defined as the art of hiding secret text or data undercover in media, where it is not possible to know this secret text is hidden while the text is transmitted through the internet. Embedding secret messages in digital (audio, video and image) files is the most challenging task in steganography because the human auditory system (HAS) and human visionary systems (HVS) have such dynamic ranges over which they can observe (Nehru and Dhar 402-06). The only weakness in HAS and HVS involves distinguishing the differences from the original data. This weakness is what must be exploited to encode secret messages without the secret message being detected (Nehru and Dhar 402-06). There are different types of steganography [10]:

1) Pure steganography

2) Secret key steganography
3) Public key steganography

Let’s discuss in depth about different types of steganography methods below:

1) Pure steganography

This is one of the basic types of steganography that hides the data in digital files without any key [10]. Here, the lack of key functionality degrades the level of security for the hidden data, as it can be easily decrypted. Figure 2 displays the concept of pure steganography.

![Figure 2. Pure Steganography [10]](image)

2) Secret key steganography

Another type of steganography is similar to pure steganography, but it adds the extra feature of key functionality. A key is provided at the time of hiding the message and the same key needs to be used to extract the message. This concept provides better security than other steganography types. The major focus of this paper will be on secret key steganography.

3) Public key steganography

This type of steganography technique is similar to secret key steganography, but it uses the concept of a public key and a private key. Here, the private key is used to hide the
message in digital files and the public key is used to extract the secret message from the
digital files.

**Uses of Steganography**

Steganography is necessary to hide data from unauthorized users, particularly in the
following areas:

- **Governments:** There are many governments and investigative agencies that need
to hide sensitive information when sending or receiving the data from their
officers.

- **Businesses:** The competition among companies in the same field is growing daily. Each competitor wants to overcome its opponents by any means, so securing confidential data is very important. In this case, while sending or receiving the information, they use steganography.

- **Individuals:** Every individual has some private data to store, and they don’t want anyone to view this information without permission, so the need for steganography arises when sending or receiving this information.

**1.3 Characteristics of Steganography**

Nowadays, sending and receiving secret messages through the internet without proper security is not safe. Steganography overcomes this problem by hiding the information in digital files. This steganography concept helps to restrict unauthorized access to private or secure information. The characteristics of steganography are as follows:

- **Confidentiality:** It is harder for an intruder to detect the existence of the hidden data in the digital files.
• **Survivability**: This means verifying that the data is not lost or destroyed while in transmission.

• **No detection**: The intruders cannot detect the secret message that is hidden in a particular digital file.

1.4 **Reasons for Steganography**

Personal and private information, for example, e-commerce transactions, highly secured data of a company, etc., can be kept secret from unauthorized users by using steganography. Steganography therefore can play a major role in hiding this secret data from intruders by restricting the misuse of data by unauthorized users.

1.5 **Algorithms Used for Hiding Data**

There are many methods used to hide text in audio, video and image files. One of the most common methods is least significant bit.

1.5.1 Least Significant Bit

Least significant bit (LSB) coding is one of the simplest ways to embed information in a multimedia file. The example figure of LSB is shown below (see figure 2).

![Figure 3 Example of LSB](image)
1.6 Objectives

The objective of this project is to hide the existence of secret text in digital files from everyone, except the sender and receiver. The objective of this project can be achieved by using the least significant bit (LSB) algorithm for hiding text in digital files. Then the sender and receiver can use this tool to hide and extract the secret text in digital files.
2. Previous Research, Motivation, Scope, and Proposed Research

2.1 Previous Research

The concept of steganography was first used by Herodotus in 440 BC to notify Greece about the invasion from Persia. He wrote his message on a tablet and covered the message with wax. Nowadays, the majority of messages are hidden within digital files of images, audio, and video.

The list of the contributors to the new technology is long. Mehboob and Faruqui (1-5) proposed a technique to hide data in colorful images using LSB. Yadav, Mishra, and Sharma (1-5) proposed video steganography based on the LSB technique, and Nehru and Dhar (402-06) proposed audio steganography based on LSB approach. Sheshidhar Odeti (8) developed a steganography tool that implements audio steganography or .wav steganography. This tool concentrates on hiding the secret messages only in audio files such as .wav, mp3, wma, and wmv. Rakesh Kumar Srirangam (9) developed a novel steganography tool that can be used to encrypt and decrypt an image. This tool is implemented by using the concepts of cryptography and steganography, which then achieves multiple levels of security. This tool hides secret messages only in images. Anudeep R. Kandi (10) developed a tool to demonstrate common data hiding techniques to help computer forensics instructors. He proposed a hybrid model ENHARM that provides multiple security for text by MD5, encrypting and decrypting and even using randomly generated 6*6 matrixes. Finally, Preethi Chittimalla (11) proposed a PowerPoint file steganography that is one of the new approaches in which data hiding is done in sound/custom animation effects of the PowerPoint file.
2.2 Motivation:

As technology expanded, users started migrating information into digital form. It is difficult to secure these digital data from hackers while sending or receiving files from another person over the internet. So we need some secure method to communicate that’s the main reason for developing this tool. This proposed tool will eliminates user effort of using different tools.

2.3 Scope

The primary idea for developing this project is to protect confidential information from unauthorized users. A new tool is proposed here with the implementation of the steganography concept. At the time of hiding, user provides a key file, a carrier file, and a secret text as input and which produces a stego-digital file as output. At the time of extraction, the receiver inputs the generated stego-digital file and the same key file as input and gets the secret message as output. The project was tested with various inputs to ensure that the generated stego-digital file had no date loss.

2.4 Proposed Work

This paper proposes a new tool that hides the secret text in audio, video, and images using LSB. I am also working on a tool to hide secret text files in .midi, .dll, and .exe files irrespective of result.

The proposed tool supports different carrier files they are

a) Audio – .wav, .mid.

b) Video – .avi.

c) Images- .bmp, .tif, .tiff, .png.
3. Architecture

Figure 4 describes the architecture of the proposed tool’s implementation. The modules of the steganography tool are also included in the architecture. The user can be either the sender or the receiver. The cover file in the above architecture can be image, audio, and video. The in-depth flow of this architecture is described in the design approach.

Figure 4. Architecture of Proposed Steganography
4. Design Approach

As shown in Figures 5 and 6 for this algorithm, there are four main steps: (1) alteration, (2) modification, (3) verification, and (4) reconstruction of the sample.

1. Alteration

In the first step, the target bits will be replaced with the message bits in the cover file. The bits wanted to alter are placed in a layer to become the target bits. This is a simple substitution (Nehru and Dhar 402-06).

2. Modification

Desired results depend on this step. Here, we use intelligent and more efficient algorithms. The objective of using algorithms in this step is to decrease the error rate and improve the transparency. Because the transparency is simple, we can adjust and modify the differences between the original and modified cover files by using more intelligent algorithms, which can improve the transparency (Nehru and Dhar 402-06).

3. Verification

This stage does not involve the use of algorithms because the algorithm implementation was done in the previous step. Now, we need to verify the quality of the outcome. If the difference between the original cover file and new cover file is acceptable and minimal, then we accept this cover file, otherwise we reject it and begin the process again (Nehru and Dhar 402-06).

4. Reconstruction of Sample

This last step results in the creation of the steganography file. There are two states of input: either the original file is input or the modified file is the same. In truth, the
algorithm does not alter all the files or the predictable files. Modification of these files depends on the decision of intelligent algorithms and their environments.

4.1 Hiding Secret Messages in Digital Files

Figure 5 describes hiding a secret file in a cover file, we began by selecting a key file and an acceptable cover file. The tool alters and modifies the bits of the cover image to allow the insertion of the secret message in the cover image. After this insertion is completed, a new, acceptable file is generated. This new file is called a stego file.

The cover file is any file we use for saving a secret text message. This cover file can be an audio, video, or image. The secret file is a file that we want to hide under the cover file.
when sending or receiving from another user over the internet. In this paper, our focus is on hiding a text file. A key file is used as the key to hide a message. The same key should be used to extract the message. The key file can be a text, audio, video, and image. The stego file combines the secret message and the cover file. After the secret message is hidden in the cover file, a file is generated with the new name of this stego file.

### 4.2 Secret Message Extraction

Figure 6 shows the process of extracting the secret message from the stego file. To extract the secret message, we need the same key file we used to hide the message. We begin by verifying that key file. After verification is successful, the tool extracts the secret message from the cover file.

![Diagram](image.png)

Figure 6. Extracting a Secret Message from a Digital File (Nehru and Dhar 402-06)
4.3 Software Requirements

The software technologies that are used for the project development are as follows:

Table 1 Software Requirements

<table>
<thead>
<tr>
<th>Category</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Windows 7</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>Visual Studio 2010</td>
</tr>
<tr>
<td>Web technology</td>
<td>ASP.NET</td>
</tr>
<tr>
<td>Programming language</td>
<td>C#</td>
</tr>
</tbody>
</table>

4.4 Unified Modeling Language (UML) Diagrams

UML is a standardized modeling language enabling developers to specify, visualize, constructs and document artifacts of a software system. The UML diagrams are divided into two categories:

a) Structure Diagrams

b) Behavior Diagrams

4.4.1 Class Diagram:

Figure 7 describes the class diagram it is a type of structure diagram that shows the structure of the classes, attributes, operations and relationship among them. Here the class diagram describes about the file utility and file type.
4.4.2 Use Case Diagram

In Figure 8, the user can be a sender or receiver. The user can perform image, audio, video, and library steganography. The user can interact with the system using the interface of the tool.
4.4.3 Sequence Diagram

4.4.3.1 Image

Figure 9 is the sequence diagram showing the flow of execution for an image. The user interacts with the system to hide the secret text message in the cover image file. The user performs the required system inputs that produce the output.

![Sequence Diagram for Image](image)

Figure 9. Sequence Diagram for Image

4.4.3.2 Audio

Figure 10 is the sequence diagram showing the flow of execution for audio. The user interacts with the system to hide the secret text message in the cover audio file. The user performs the required systems inputs that produce the output.
4.4.3.3 Video

Figure 11 is the sequence diagram showing the flow of execution for video. The user interacts with the system to hide the secret text message in a cover video file. The user performs the required systems inputs that produce the output.
4.4.4 Component Diagram

In Figure 12, the system consists of five main components: mainframe, image steganography, audio steganography, video steganography, and library steganography. All of these components can be performed individually without interlinking.

![Component Diagram](image)

Figure 12. Component Diagram

4.4.5 Collaboration Diagram

4.4.5.1 Image

Figure 13 describes about the collaboration diagram for image. This diagram clearly describes how the user interacts with all the objects to perform the required task.
4.4.5.2 Audio

Figure 14 describes about the collaboration diagram for audio. This diagram clearly describes how the user interacts with all the objects to perform the required task.
4.4.5.3 Video

Figure 15 describes about the collaboration diagram for video. This diagram clearly describes how the user interacts with all the objects to perform the required task.

Figure 15. Collaboration Diagram for Video

4.4.6 Deployment Diagram

Figure 16 describes about the deployment diagram. This diagram describes about the requirements that the tool need to build the tool.
Figure 16. Deployment Diagram
5. Implementation

Implementation of this hiding data tool consists of two modules:

1) Hiding secret data in digital files
2) Extracting secret data from digital files

5.1 Hiding secret data in digital files

In this module, we mainly focus on hiding the secret text message in digital files, such as image, audio, video. Below, the implement of the process for this module is discussed.

Figure 17 represents the home page of the tool for hiding secret data in digital files. Here, we need a key file, a supported digital file, and the secret data that we need to hide.

Figure 17. Home Screen for Hiding Secret Data in Digital Files
In the first step, from Figure 18, a key file needs to be added to the tool. As mentioned, the key file can be the text, audio, video or image. Let’s consider the key file as text.

Figure 18. Key File

Figure 19. Adding Digital File
In Figure 20, the image is considered a digital file. In the filename column, added a digital file and in save as column give it a name to save the generated stego file. In addition it has the option to change the carrier units and create extra-noise. After this is done, click on the save button, which leads to home page if everything is working fine or else an error is displayed. The error can be due to file reading problem, insufficient file size, or an unsupported format.

![Figure 20. Browsing and Naming a File](image)

In the last step, we selected the secret file with the secret information and hide it by clicking on the hide message button. If everything is perfect, then the message shown in Figure 21 pops up saying hiding data is completed.
5.1.1 Hiding Text in Images:

Figure 22 describes the flow diagram for embedding text in images. First, the user needs to select a key file, a carrier file (image), a secret message, and a stego image. Then, the user gives all of these inputs for validation and checking whether there are any exceptions. If no exceptions are found, the software makes the required alterations and modifications and generates a stego image file. This file is the combination of the secret message and carrier image. If any exceptions are found, the process is repeated.
5.1.2 Hiding Text in Audio (Wav, Midi)

Figure 23 describes the flow diagram for embedding text in audio. First, the user needs to select a key file, a carrier file (audio), a secret message, and a stego audio. Then, the user gives all of these inputs for validation and checking whether there are any exceptions. If no exceptions are found, the software makes the required alterations and modifications and generates a stego audio file. This file is the combination of the secret message and carrier audio. If any exceptions are found, the process is repeated.
5.1.3 Hiding Text in Video:

Figure 24 describes the flow diagram for embedding text in video. First, the user needs to select a key file, a carrier file (video), a secret message, and a stego video. Then, the user gives all of these inputs for validation and checking whether there are any exceptions. If no exceptions are found, the software makes the required alterations and modifications and generates a stego video file. This file is the combination of the secret message and carrier video. If any exceptions are found, the process is repeated.
Figure 24. Flow Diagram for Embedding Text in video
5.2 Extracting the Secret Text in Digital Files

In this module, the major focus is on extracting the secret text message from the digital image, audio, and video files.

Figure 25 displays the home screen for extracting the secret image. The key file should be the same at the time of hiding and extracting. If the key file is different, an error is displayed.

Figure 25. Home Screen for Extraction

In Figure 26, clicking on the add button adds the stego file generated at the time of hiding. This stego file is a combination of the original file and the secret text.
Figure 26. Adding a Stego File

Figure 27 displays the last step for the extraction process. After the stego file is added, then users need to click on the extract message button. If the extraction process is successful, then the message popup says that the extraction is complete. The secret text can either be displayed on the screen or saved on a computer.
5.2.1 Extracting Text from Image:

Figure 28 describes the flow diagram for retrieving text from image. The user needs to input the key file and generated stego image. The same key file should be used at the time of hiding and extracting. After the input is provided, the data is validated and checked for exceptions, and if any exceptions are found, the process begins again. If no exceptions are found, then an alteration is done to extract the secret message from the image file. This extracted secret message will then be displayed on the screen or else be saved in a file.
5.2.2 Extracting Text from Audio (Wav, Midi):

Figure 29 describes the flow diagram for retrieving text from audio. The user needs to input the key file and generated stego audio. The same key file should be used at the time of hiding and extracting. After the input is provided, the data is validated and checked for exceptions, and if any exceptions are found, the process begins again. If no exceptions are found, then an alteration is done to extract the secret message from the audio file. This extracted secret message will then be displayed on the screen or else be saved in a file.
5.2.3 Extracting Text from Video:

Figure 30 describes the flow diagram for retrieving text from video. The user needs to input the key file and generated stego video. The same key file should be used at the time of hiding and extracting. After the input is provided, the data is validated and checked for exceptions, and if any exceptions are found, the process begins again. If no exceptions are found, then an alteration is done to extract the secret message from the video file. This extracted secret message will then be displayed on the screen or else be saved in a file.
Figure 30. Retrieving Text from Video
6. Testing and Results

Testing is one of the efficient processes to find errors during a program’s execution. During this process, we can find different kinds of errors and correct them in order to make the tool effective. This tool was executed and tested to identify any errors. Implementation of this tool was divided into modules and we test this tool in every module (user interface) to make sure it works properly without errors.

Test Case 1

To check whether the users could view the hiding and extracting phases, when a user clicked the hide or extract button in the proposed tool, then the following was observed:

- The users’ view hiding phase is when the hide button is clicked.
- The users’ view extracting phase is when the extract button is clicked.

Figure 31 describes the home page for what is hidden. When the hide button is clicked, it shows the home screen of that page where users need to enter the required inputs that, in turn, generate the stego output.

![Figure 31 Checking Hide Button](image-url)
Figure 32 describes the home screen of the extraction page. Clicking the extract button leads to the extracting page where the stego file is provided as input and the secret message is extracted from that stego file.

![Figure 32. Checking the Extract Button](image)

**Test Case 2**

In this case, the options “Browse” and “Text,” along with the “+” and “-” buttons are tested as working properly or not.

- In the Text option, is the typed input correctly considered or not?
- In the Browse option, can correct file be found?
- The + button tests whether users can successfully add digital files.
- Md5 hashing for secret message.
To check this text option in the tool, Figure 33 shows that the text option is working properly.

![Figure 33. Checking Text Options](image)

In this case, the point is to check whether the browse option works properly. As shown in Figure 34, it is.

![Figure 34. Checking the Browse Option](image)
In testing whether the + button is working properly or not. Figure 35 show that the + button adds digital files successfully.

Figure 35. Testing + Button

Figure 36 successfully tests that the –button works successfully. This button can successfully delete the selected option.
Figure 36. Testing the –Button

Figure 37 successfully tested the cancel option. After clicking the cross mark as shown in Figure 37, the users is directed to the previous page and existing written data is canceled.

Figure 37. Testing the Cancel Option
Test Case 3

In this test case, the focus is on hiding and extracting the text from digital (audio, video and image) files. The main idea is to check whether we can successfully hide and extract the text in digital (audio, video and image) files. In addition, we checked by changing the noise and carrier capacities.

Figure 38 shows that the required inputs were provided to hide the secret texts in digital files. The inputs are key file, secret message, and cover file. We changed some of the carrier units and created some extra noise.

Figure 38. Inputs for Hiding Text in Digital Files

Figure 39 shows that we can successfully hide secret texts in digital files and produce a stego-file. The stego-file is a combination of the cover file and secret message.
Figure 39. Output Screen for Hiding Text in Digital Files

Figure 40 shows the required inputs for extracting a secret message. These inputs can be the same key file used at the time of hiding the generated stego-file.

Figure 40. Inputs for Extraction of a Secret Message from a Stego-File
Figure 41 shows the results of a completely extraction. In this final stage, the secret message is extracted successfully from the generated stego file, which shows that this tool is working properly for all digital (audio, video and images) files.

![Output Screen for Extraction the Secret Message from Stego-File](image)

**Figure 41. Output Screen for Extraction the Secret Message from Stego-File**

**Test Case 4:**

In this test case the focus is on the secret message. So we will use hashing functionality for secret message before hiding and after extraction. If the hashing is different for both the files, then the secret message is successfully hiding and extracting from the cover file. Here md5 hashing is used to check the hashing of the document.

In figure 42 describes two files with hashing value. One file is the secret message before hiding and other file is the secret message after extraction. As both the hashing values are
different we can state that secret message is not a temporary file or cache file. This secret
message is successfully extracted from the cover file.

Figure 42. Hashing the Secret Message
7. Conclusions

This research introduces a new tool. This newly developed steganography tool is used for embedding and de-embedding digital files. For this project, confidential data was secured by using the least significant bit (LSB) algorithm. LSB coding is one of the simplest ways to embed information in digital files as it replaces LSB with the message to be encoded. The secret message is embedded into a digital file with a key file that generates a stego file. The digital file can be image, audio, or video, and a stego-file combines the secret message and digital file. The key file is common for hiding and extracting the secret message. At the time of extraction, we gave the stego file and key file, both of which de-embedded and displayed the secret message. In general, this tool allows users to hide secret text in digital files and extract the same secret text from those digital files. The tool can be used to hide a secret message from hackers while transferring the message among users. A user with some basic knowledge can use this tool either as a sender or as a receiver. Some test cases are performed for both hiding and extracting the secret message in image, audio, and video. This demonstrates that using this tool can achieve the confidentiality of secret data.
8. Future Work

To add additional security, the future focus to on developing this tool to work in all formats, i.e., hiding digital files in any digital file, as this tool focuses only on hiding text in digital files. In addition, tool also gets more security when both the steganography and cryptography concepts are used together.
BIBLIOGRAPHY AND REFERENCES


10) Kandi, Anudeep R. DESIGN AND IMPLEMENTATION OF A TOOL TO HELP COMPUTER FORENSICS INSTRUCTOR TO DEMONSTRATE COMMON DATA HIDING TECHNIQUES. Diss. Texas A&M University, 2012.