D Y PATIL INTERNATIONAL UNIVERSITY
AKURDI PUNE

StegoStream : Securing Multimedia Data Using Steganography

By

Pranil Anil Choudhari

PRN:20220804036

Pranav Kivade

PRN:20220804051

Under Guidance

Of

Sarika Jadhav

Submitted to

School of Computer Science, Engineering and Application

In partial fulfilment of the requirements

For the award of the degree

Master of Computer Applications (MCA)

2022-2023

D Y PATIL INTERNATIONAL UNIVERSITY
AKURDI PUNE

School of Computer Science, Engineering and Applications

CERTIFICATE OF COMPLETION

This is to certify that the project report entitled **STEGOSTREAM : Securing Multimedia Data Using Steganography** submitted to **School of Computer Science, Engineering and Application,** ***D Y Patil International University*** in partial fulfilment of the requirements for the ***Project- I course,*** **SEM II** of the degree of **Master of Computer Applications (MCA),** is an original work carried out by **Mr. Pranil Anil Choudhari** and **Mr. Pranav Rajesh Kivade** with PRN **20220804036** and **20220804051** under my guidance.The matter embodied in this project is genuine work done by the student andhas not been submitted whether to this university or to any other university for the fulfilment of the requirements of any course of study.

_____
Signature of the student:
Date: _____
Name of the student: _____
PRN: _____

_____
Signature of the Guide:
Date:_____
Name of the **Guide**: _____
Name of the School: _____

_____
Signature of the project In-charge:
Date: _____
Name of the **project In-charge**: _____
Name of the School: _____

_____
Signature of the HoD:
Date: _____
Name of the **HoD**: _____
Name of the School: _____

_____
Signature of the Director:
Date:_____
Name of the **Director**: _____
Name of the School: _____

ACKNOWLEDGEMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed guide **Ms.Sarika Jadhav (Organization),** for her valuable guidance, encouragement, and help in completing this work. Her useful suggestions for this whole work and cooperative behaviour are sincerely acknowledged. I would like to express my sincere thanks to the project in-charge **Ms.Sarika Jadhav** and Head of the department **Dr.Maheshwari Biradar** for giving me this opportunity to undertake this project. Also wish to express my gratitude to the Director **Dr.Bahubali Shiragapur** for his kind-hearted support.

I am also grateful to my faculty members for their constant support and guidance. I also wish to express my indebtedness to my parents as well as my family member whose blessings and support always helped me to face the challenges ahead. The end I would like to express my sincere thanks to all my friends and others who helped me directly or indirectly during this project work.

Place: Student Name:

Date: PRN:

ABSTRACT:

In today's increasingly digital world, data privacy and security have become pressing concerns. As individuals and organizations continue to rely more heavily on digital media to communicate, store data, and conduct business, it becomes more critical to protect the confidentiality and integrity of this information. This is where steganography comes into play.

Steganography is the technique of hiding secret data within an ordinary, non-secret, cover media in a way that does not allow detection of the hidden data.

This will increase the security of the Organization. Using our steganography tool , one can hide and secretly send the data (such as Text , PDF and Word file) by hiding it behind an Image , Audio or Video . steganography can be a useful tool for adding an extra layer of security to digital communications and data storage. However, it should be used in conjunction with other security measures, such as encryption and proper network security practices, to ensure the highest level of data privacy and protection.

1. INTRODUCTION:

1.1. Introduction :

Steganography is a term that originated from two Greek words, "Steganos" meaning "covered," and "Graptos" meaning "writing." It refers to the practice of concealing data and information in plain sight. In today's world, the sharing of data and information has become increasingly common, and the security of this information has become essential with the development of electronic information and data sharing devices. Steganography and cryptography are two methods available for information and data security, but people often confuse the two. Although both methods are used for data and information hiding, they differ in their approach. Steganography is a way of hiding data and information such that a person is unaware that hidden data or information is present. It exploits human perception, and the hidden data or information is not visible directly. Steganography is typically used to hide files inside other files, while cryptography is a way of protecting data that can be known through decryption. In cryptography, the message is given in an encrypted form, and there is an encryption key that is shared between authorized persons. Anyone with this encryption key can decrypt the message and access the data and information. Cryptography does not involve the hiding of data, but rather, it is a method of protecting it. Steganography can involve different types of data, including audio, video, image, and text. The goal of steganography is to make the existence of the message as inconspicuous as possible, allowing it to be transmitted securely without being detected. This technique has been used throughout history, and today, it is commonly used in digital communications to protect sensitive information from interception and unauthorized access.

1.2. Objectives :

- Develop a system that can embed the secret message in the form of text , pdf and word file within the Image,Audio and Video without changing the visual quality or audio quality of the cover data.

- The system should be able to extract the hidden message from the image,audio & video without any loss of data.

- The embedded message should be secure and not easily detectable by attackers.

- The system should be able to handle various types of images , audio , video , pdf and message formats ,etc

- The system should have a user-friendly interface for ease of use .

1.3. Purpose :

The purpose of making a steganography project in today's date is to provide an additional layer of security to protect sensitive information in digital communication and storage. As digital security threats continue to increase, traditional security measures such as encryption and firewalls may not be enough to ensure data protection. Steganography projects enable data to be hidden in plain sight, making it difficult for attackers to even know that there is hidden data to find. This can be particularly useful in industries such as military, finance, and healthcare, where sensitive information must be protected from cyber-attacks and data breaches.

1.4. Scope :

In the coming future, with the increasing dependence on technology and the internet, the need for secure communication will only continue to grow. Steganography offers a unique solution to this problem and can be used in various domains, including military communication, banking and finance, healthcare, law enforcement, and journalism .The scope of this steganography project involves creating a tool that can securely embed and extract messages from various carrier signals while maintaining a high level of security. This project's outcome will contribute to the field of secure communication and can be further extended to explore new applications or research directions in steganography.
The project will involve designing and implementing a Python-based application that can perform steganography operations on image, audio and video files. The application should allow users to select a file and embed a message in it, as well as extract a hidden message from a file. The project should be able to handle various types of image, audio and video file formats, including but not limited to JPEG, PNG, MP3, WAV, and MP4.

1.5. Applicability:

1) **Multimedia Security:** Steganography can be used to secure multimedia content such as videos, images, and audio files from unauthorized access or tampering.

2) **Digital Watermarking:** Steganography can be used in digital watermarking, where a hidden digital signature is embedded in a multimedia file to verify its authenticity and ownership.

3) **Privacy Protection:** Steganography can be used to protect the privacy of individuals by hiding their personal information in multimedia files, making it difficult for attackers to obtain sensitive information.

4) **Forensic Investigation:** Steganography can be used in forensic investigation to detect the presence of hidden information in multimedia files and uncover any potential cybercrime or illegal activity.

5) **Political Dissent:** Steganography can be used by political dissidents to communicate securely and anonymously, protecting their identities and messages from being intercepted or traced.

6) **Steganalysis:** Steganography projects can also involve steganalysis, which is the study and development of methods to detect hidden information in multimedia files.

2. SURVEY OF TECHNOLOGIES :

To implement a full-fledged GUI steganography tool for image, video, and audio steganography using Python, several technologies can be utilized, including the Python programming language, third-party libraries such as OpenCV, NumPy,Wave, and Pillow for image processing, and GUI frameworks such as Tkinter or PyQt for creating a user-friendly interface. Additionally, the tool may utilize compression techniques and encryption algorithms such as AES, DES, RSA, or Blowfish for securing the hidden message. The combination of these technologies can enable the creation of an intuitive, efficient, and secure steganography tool for a range of multimedia file formats.

Image Steganography:

- Least Significant Bit (LSB) technique: In digital images, each pixel is represented by a binary value that indicates its color intensity. The LSB technique involves hiding the secret message by replacing the least significant bit(s) of these binary values with the secret message bits. Since the LSBs are typically not used for conveying critical information, this technique is often used for hiding data without visibly altering the image.
- Discrete Cosine Transform (DCT) technique: The DCT is a mathematical transform that converts an image from the spatial domain to the frequency domain, representing the image in terms of its frequency components. In this technique, the secret message is embedded in the high frequency coefficients of the DCT. These coefficients correspond to the edges and details of the image and are less perceptually important than the low-frequency coefficients.

Video Steganography:

- Motion Vector Technique: Motion vectors are used in video compression to represent the movement of objects between consecutive frames. This technique involves embedding the secret message by modifying the motion vectors of the video frames, while still maintaining the visual quality of the video.
- Frame Differencing Technique: In this technique, the secret message is embedded by altering the pixel values of the frames based on the differences between consecutive frames. The changes made to the frames are typically subtle and do not significantly affect the visual quality of the video.
- LSB technique: This technique involves hiding the secret message by replacing the least significant bit(s) of the pixel values of the video frames with the secret message bits. This technique is similar to the LSB technique used in image steganography.

Audio Steganography:

- Least Significant Bit (LSB) technique: In digital audio signals, each sample is represented by a binary value that indicates its amplitude. The LSB technique involves hiding the secret message by replacing the least significant bit(s) of these binary values with the secret message bits. Since the LSBs are typically not used for conveying critical information, this technique is often used for hiding data without significantly affecting the audio quality.

- Spread Spectrum Technique: This involves embedding the secret message by slightly altering the frequency components of the audio signal. This technique spreads the message bits over a broad range of frequencies and is typically used in situations where the hidden data needs to be more robust to noise and other types of signal degradation.

3. REQUIREMENTS AND ANALYSIS :

3.1. Problem Definition :

- The existing system for data encryption in audio and image files relies on traditional encryption methods, which have several limitations. These methods may not provide sufficient security against unauthorized access and can be susceptible to various attacks, including brute-force attacks and cryptanalysis.

- Furthermore, traditional encryption methods may not be suitable for scenarios where data needs to be concealed from unauthorized access completely. In such cases, steganography can provide a solution by allowing the data to be hidden within an innocuous cover file, making it difficult for unauthorized users to detect and access the hidden data.

- However, the existing steganography tools can be challenging to use and may not provide a comprehensive set of features to ensure the security and integrity of the hidden data. Moreover, these tools may require specialized knowledge and expertise, making it difficult for users with limited technical knowledge to implement steganography techniques effectively.

- Therefore, our steganography project aims to address these limitations by providing an easy-to-use, comprehensive tool that allows users to implement steganography techniques for audio and image encryption efficiently.

3.2. Requirement Specification :

3.2.1. Functional Requirements:

- File Selection: The user should be able to select an image, audio or video file from their local machine to embed or extract data.
- Data Embedding: The application should allow users to embed a message in an image, audio or video file. The message could be a text, image, audio or video file.
- Data Extraction: The application should allow users to extract hidden data from an image, audio or video file.
- Steganography Techniques: The application should employ various steganography techniques to embed and extract data in/from the selected files.
- File Formats: The application should be able to handle various file formats of image, audio and video files, including but not limited to JPEG, PNG, MP3, WAV, and MP4.

3.2.2. Non-Functional Requirements:

- Performance: The application should be able to perform steganography operations efficiently, without causing any delays or interruptions.
- Security: The application should ensure the confidentiality of the data being embedded or extracted.
- User Interface: The application should have an easy-to-use interface that is intuitive and user-friendly.
- Error Handling: The application should provide appropriate error messages and error handling mechanisms to assist the user in case of any issues or errors.
- Compatibility: The application should be compatible with various operating systems, including Windows, Linux and macOS.

]

3.3. Feasibility Study:

Our project aims to implement steganography as a means of secure communication for confidential data. This feasibility study assesses the technical, economic, and operational aspects of the project :

3.3.1. Technical Feasibility:

The technology required for implementing steganography (Such as LSB-Least Significant Bit , Spread Spectrum , Motion Vector Technique ,Frame Differencing Technique) is widely available and easily accessible. There are several open-source libraries and frameworks available for developers to use, which can help reduce development time and costs. Moreover, the existing tools and techniques for steganography have been well-documented and tested, which indicates that the project is technically feasible.

3.3.2. Economic Feasibility:

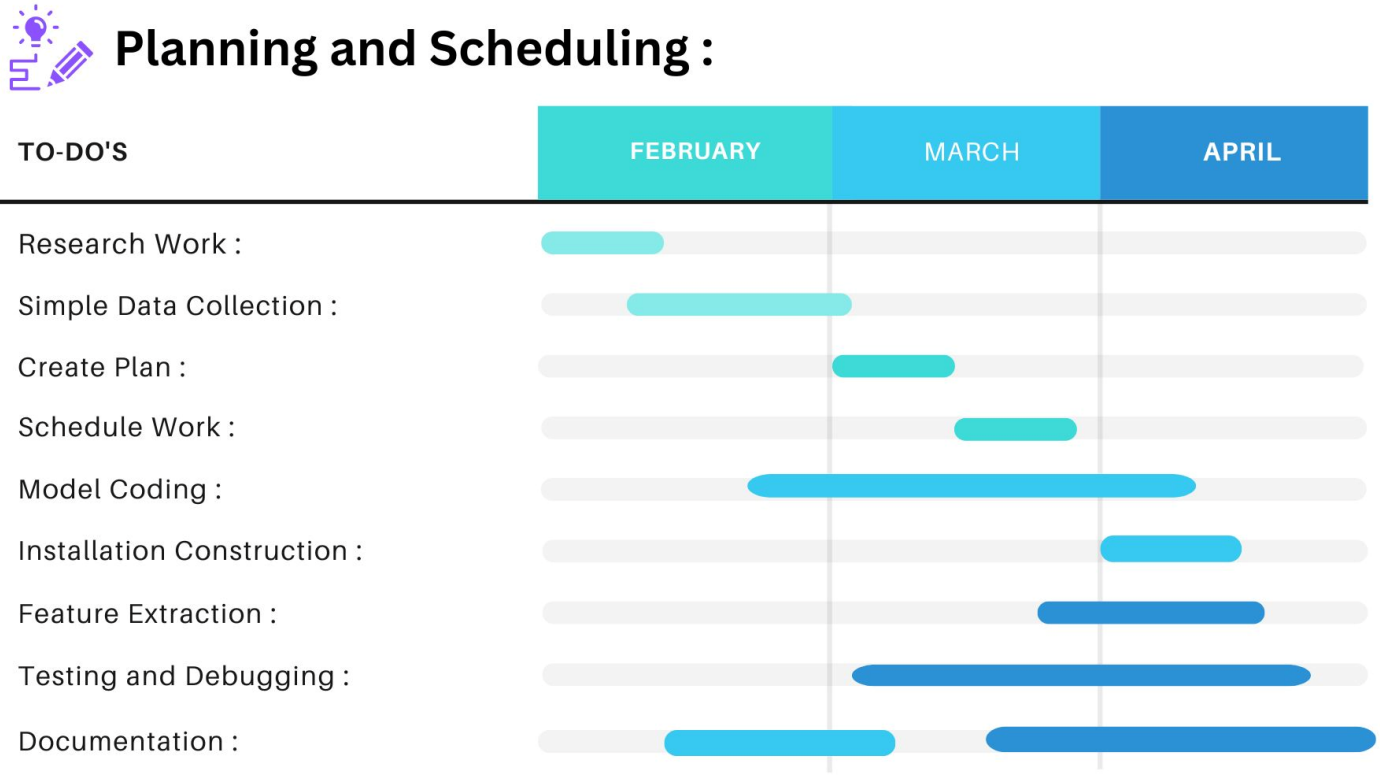
The cost of implementing steganography is relatively low as it requires only standard hardware and software tools. The project can be developed using open-source software, which is freely available. The development costs will primarily depend on the skill level of the development team and the time required to complete the project. However, the operational costs may be higher due to the need for ongoing maintenance and upgrades.

3.3.3. Operational Feasibility:

Steganography has been widely used in various domains, including military, government, and commercial applications. The project's success depends on its ease of use and the acceptance of the end-users. We need to ensure that the users are comfortable with the steganography approach, and the communication remains transparent to them. Moreover, there should be appropriate training and support available for the end-users to ensure the successful adoption of the project.

In summary, the feasibility study confirms that implementing steganography as a means of secure communication for confidential data is technically feasible, economically viable, and operationally feasible.

3.4. Planning and Scheduling :



3.5. Software and Hardware Requirements :

3.5.1. Software Requirements :

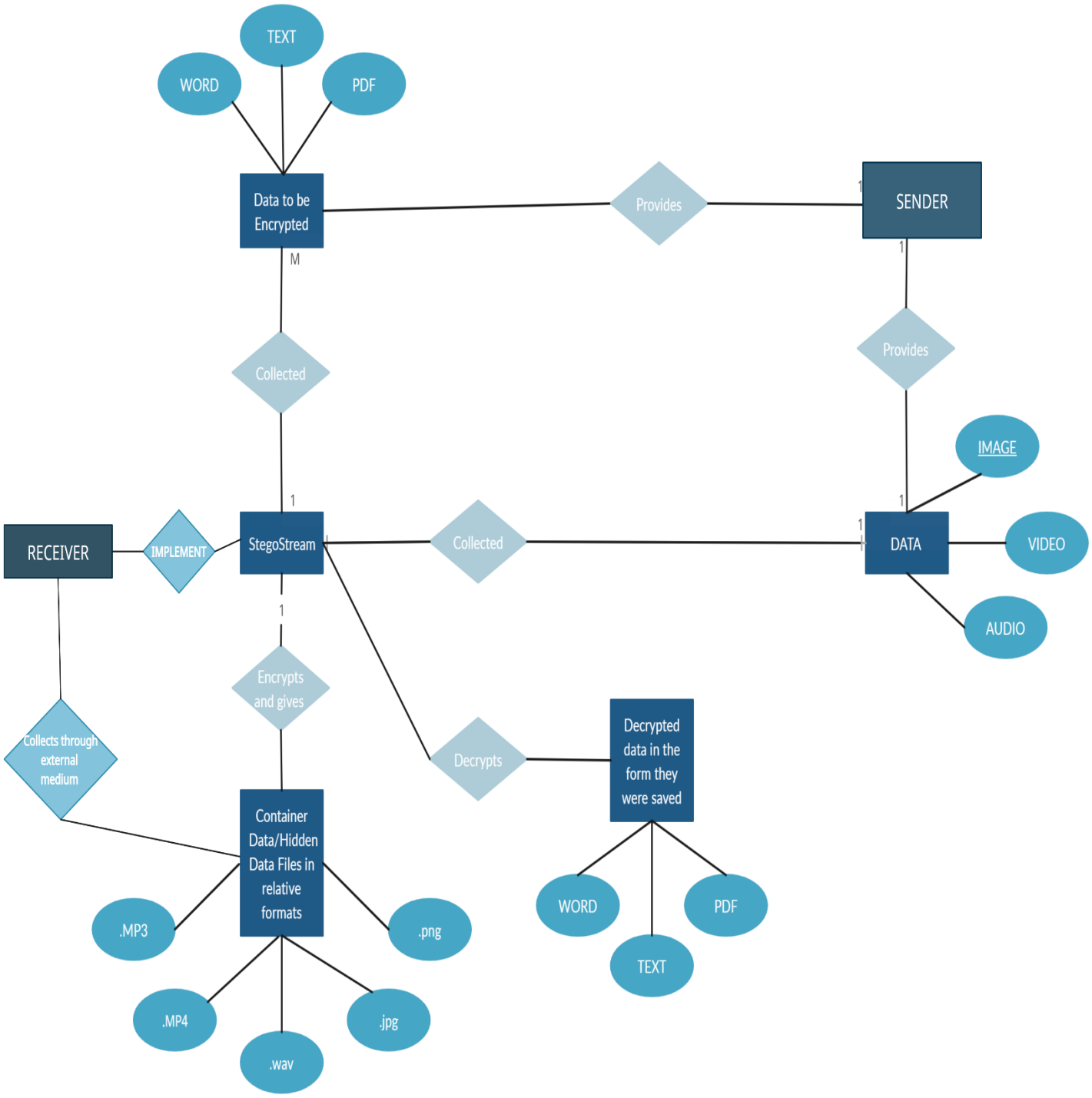
- Operating System : Windows 7 Or higher version .
- Python Preinstalled on your PC
- You need the same software on both the PCs that is , 1] PC in which the encryption occurs and 2] PC where Container Data is send and where it is Decrypted.
- PDF Viewer should have more than 1.1.0 version (which is the most basic one)

3.5.2. Hardware Requirements :

- Recommended PC specifications are Intel i3/Ryzen 3 and 4 GB RAM or more. As the system has to execute tasks and give response in least time.
- Minimum PC specifications: Pentium-pro processor or later. RAM 512MB or more
- Memory-> 1GB or above.
- HDD Space-> 50MB or above.

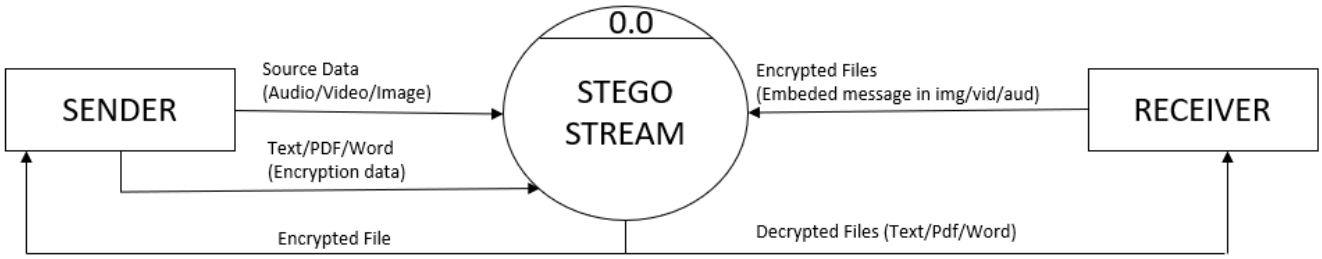
3.6. Conceptual Models / Design Documents :

3.6.1. Entity Relationship Diagram :

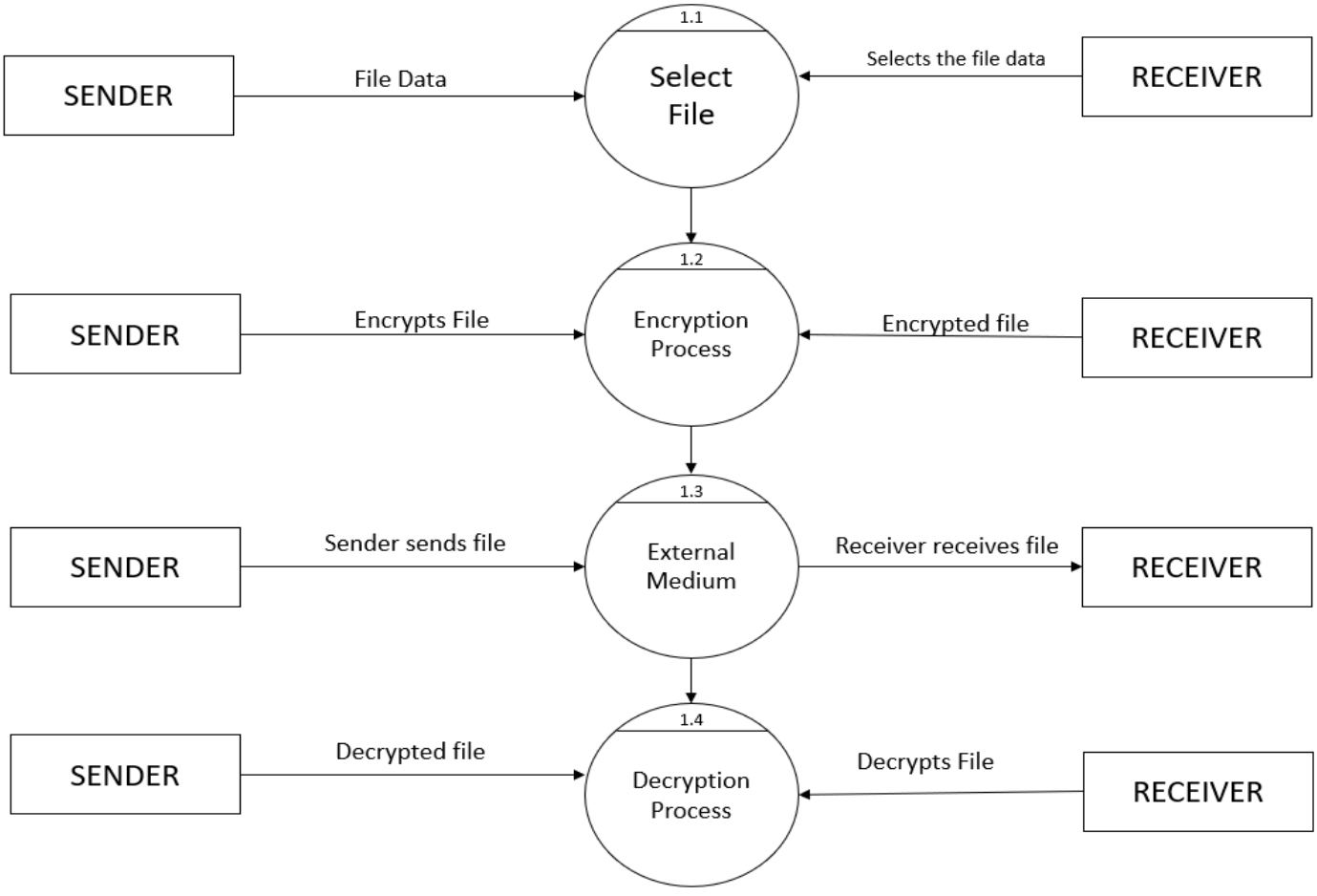


3.6.2. Data Flow Diagram (DFD) :

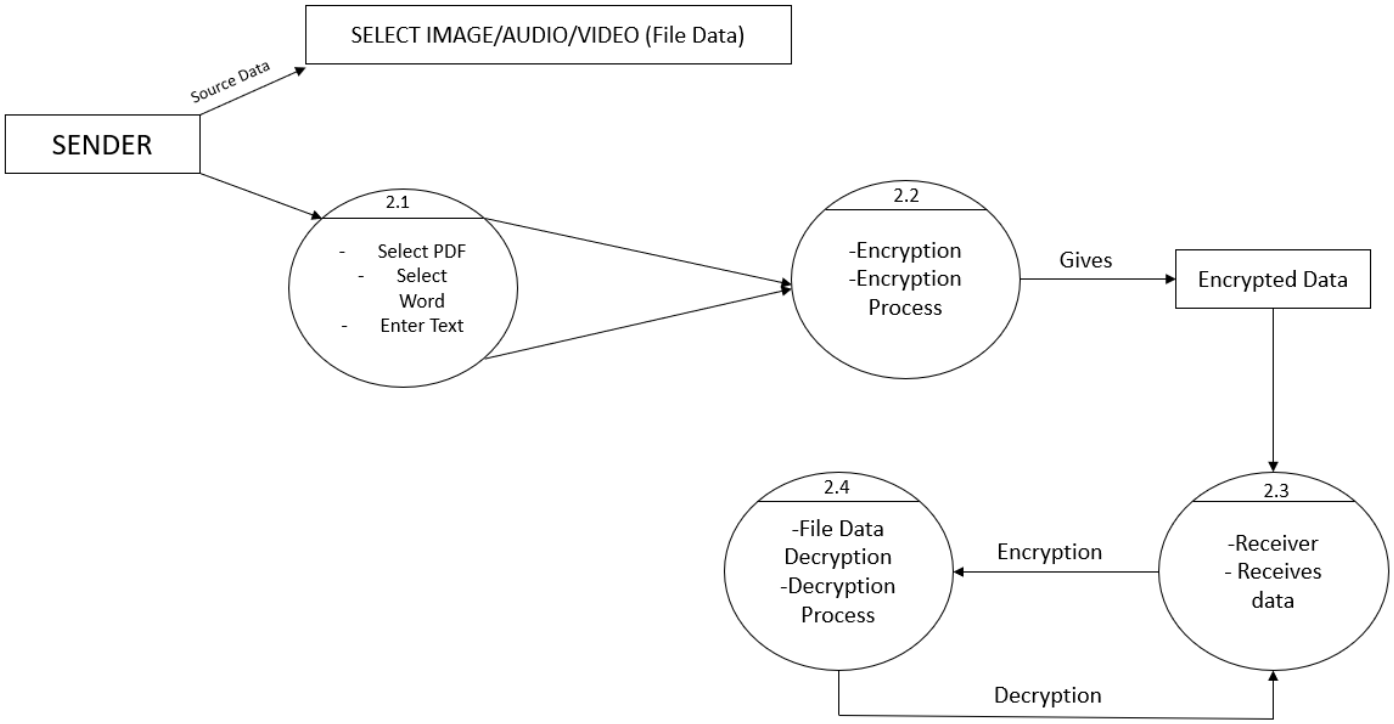
3.6.2.1. DFD LEVEL 0 :



3.6.2.2. DFD LEVEL 1 :

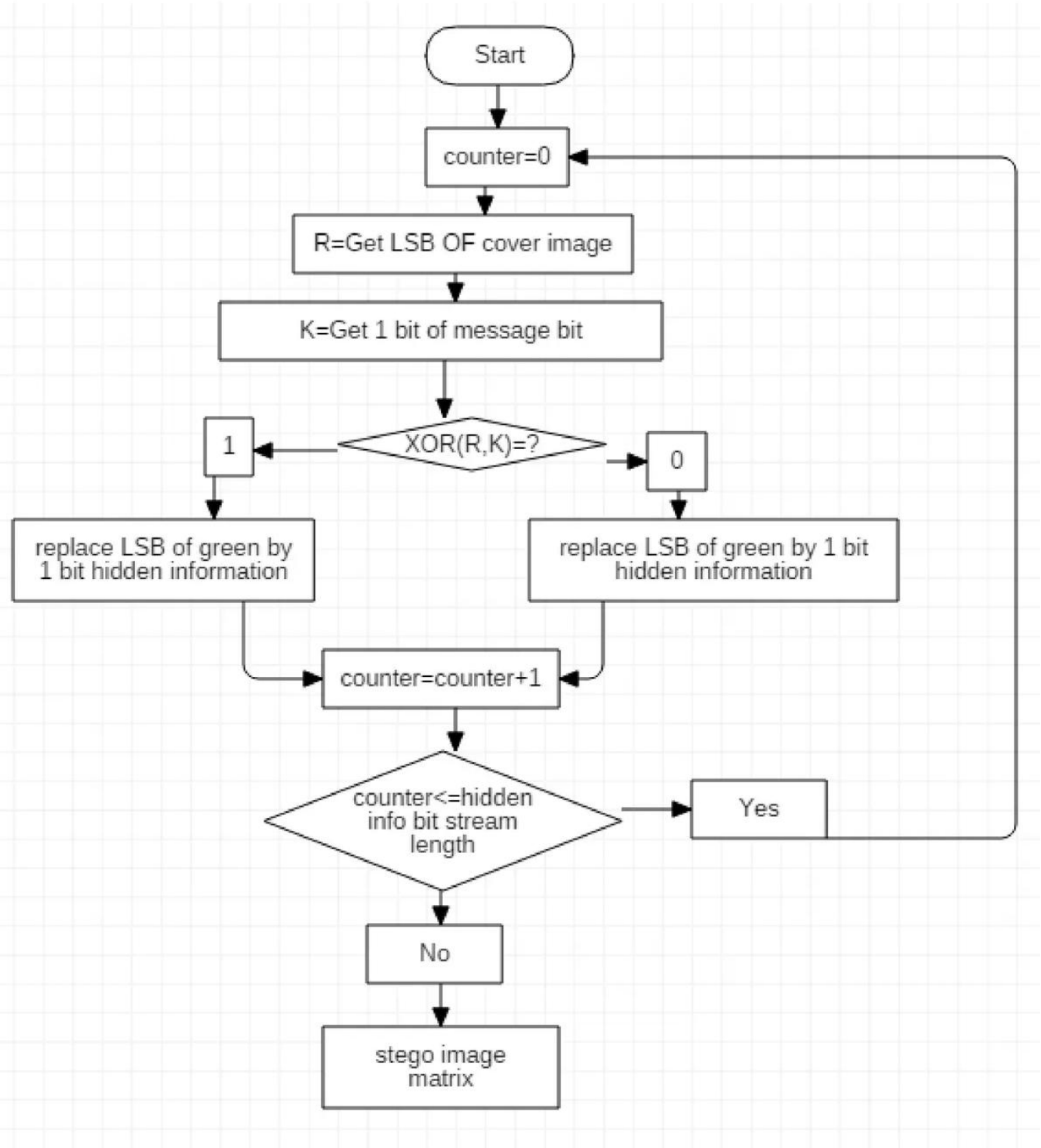


3.6.2.3. DFD LEVEL 2 :

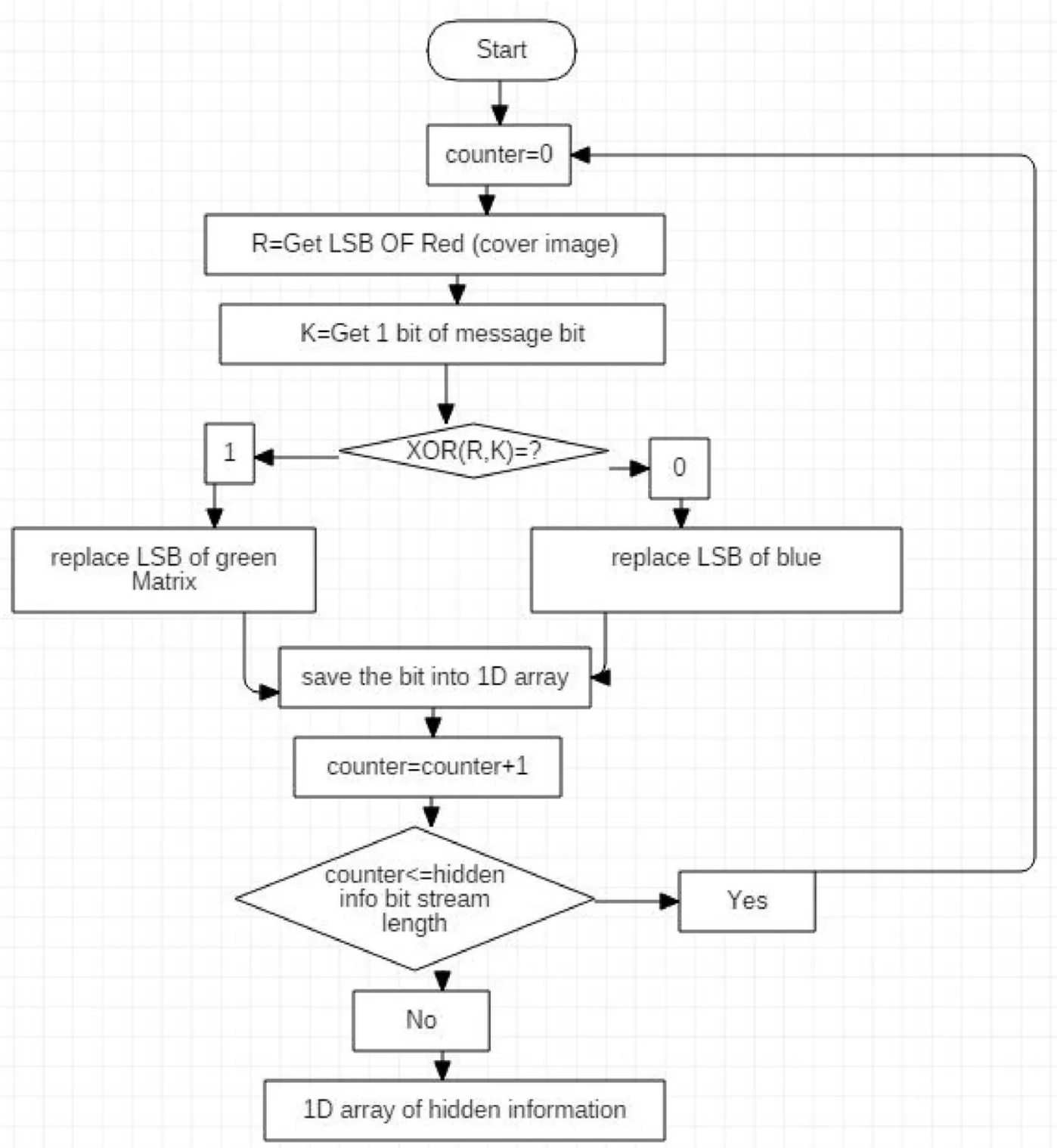


3.6.3. Flow Charts :

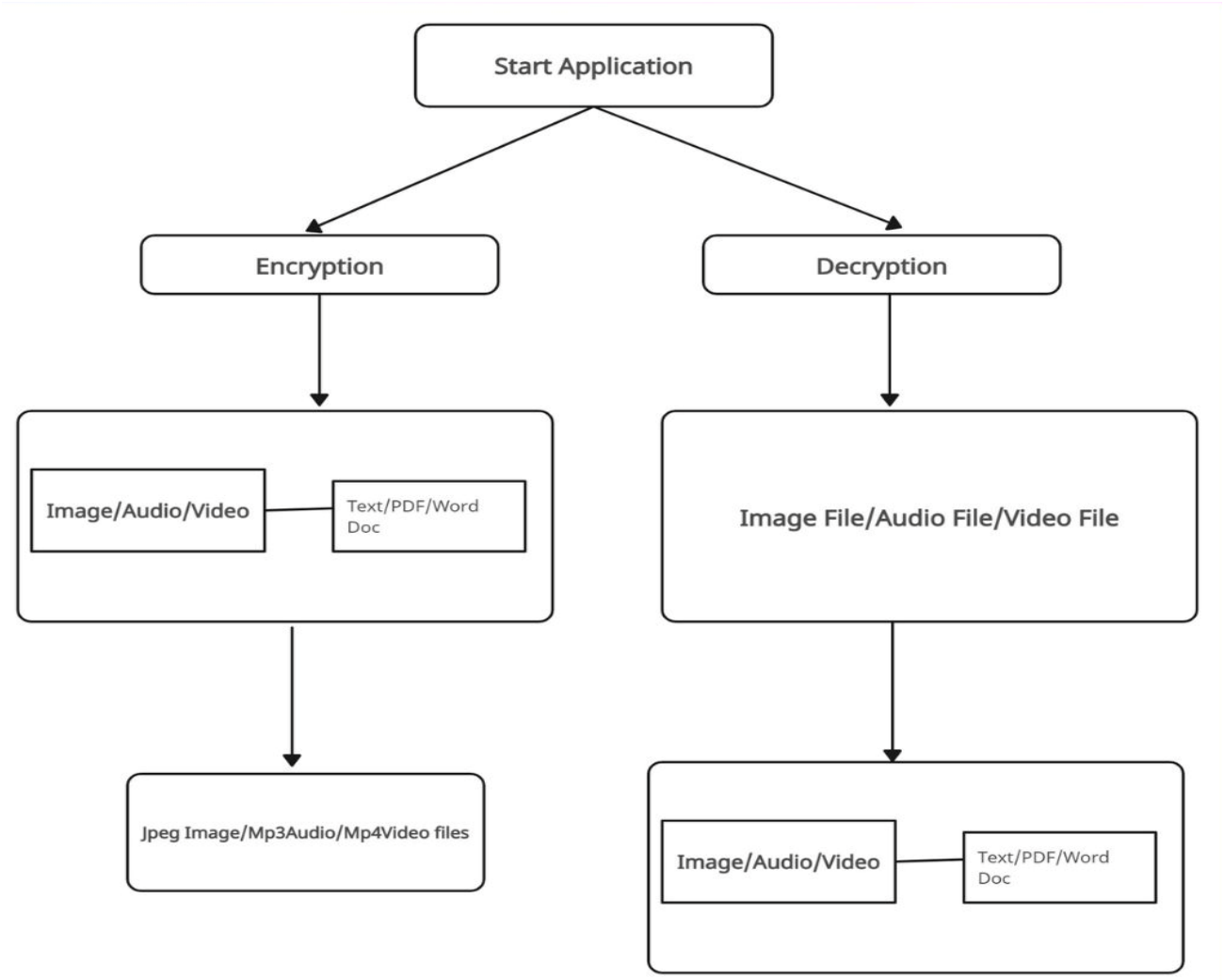
3.6.3.1. Encryption Flowchart :



3.6.3.2. Decryption Flowchart :

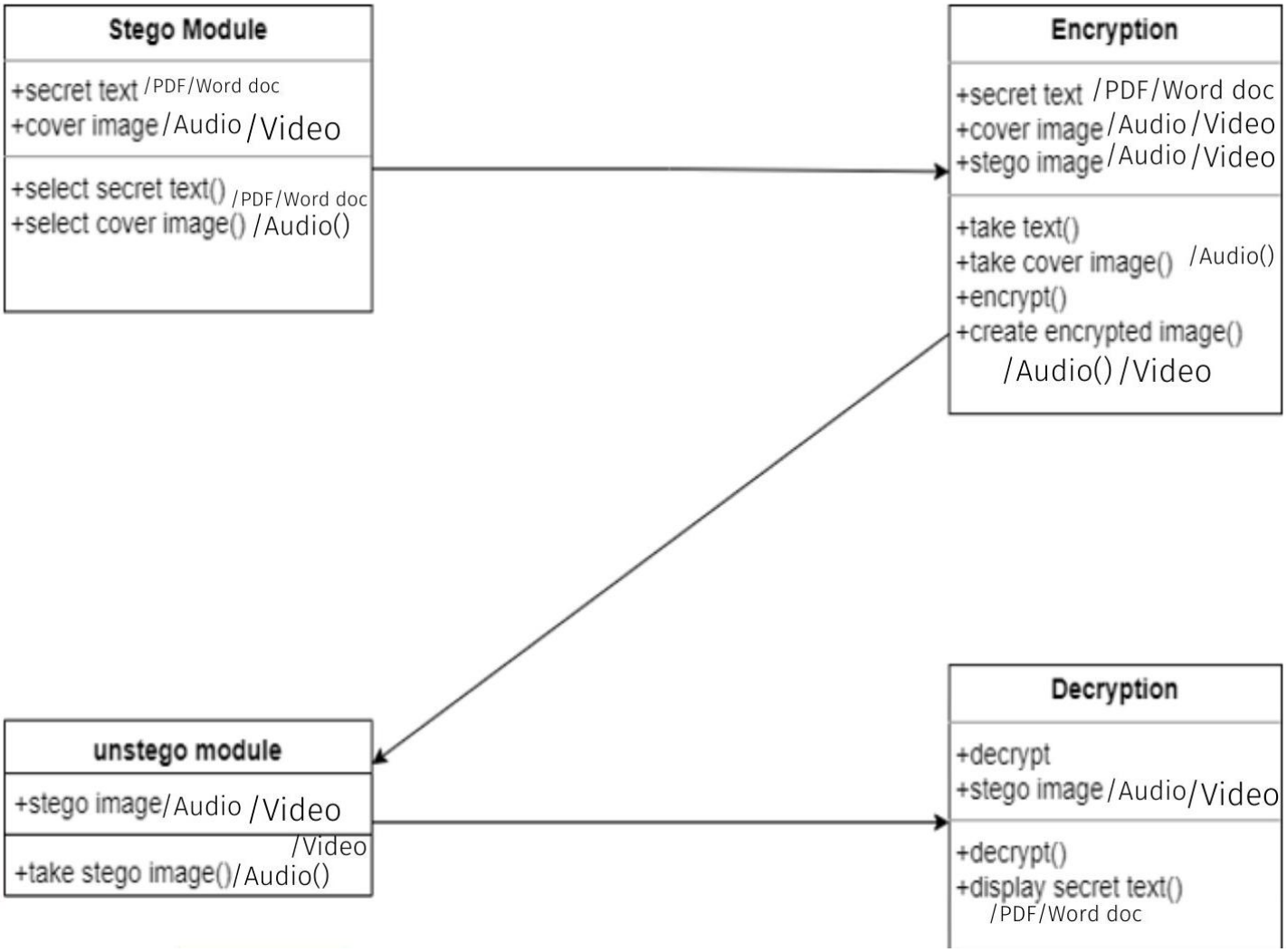


3.6.3.3. System flowchart :

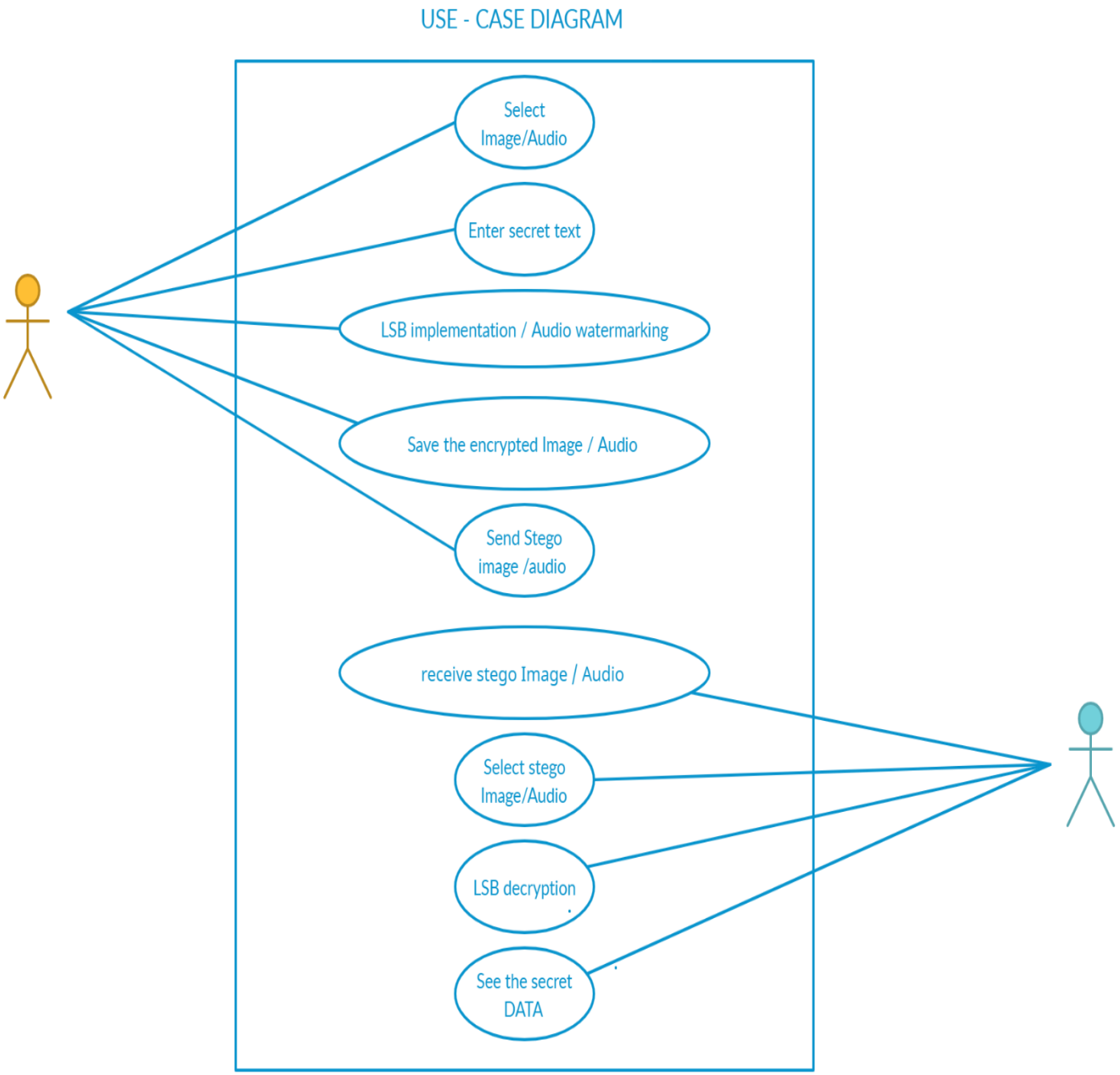


3.6.4. UML Diagrams :

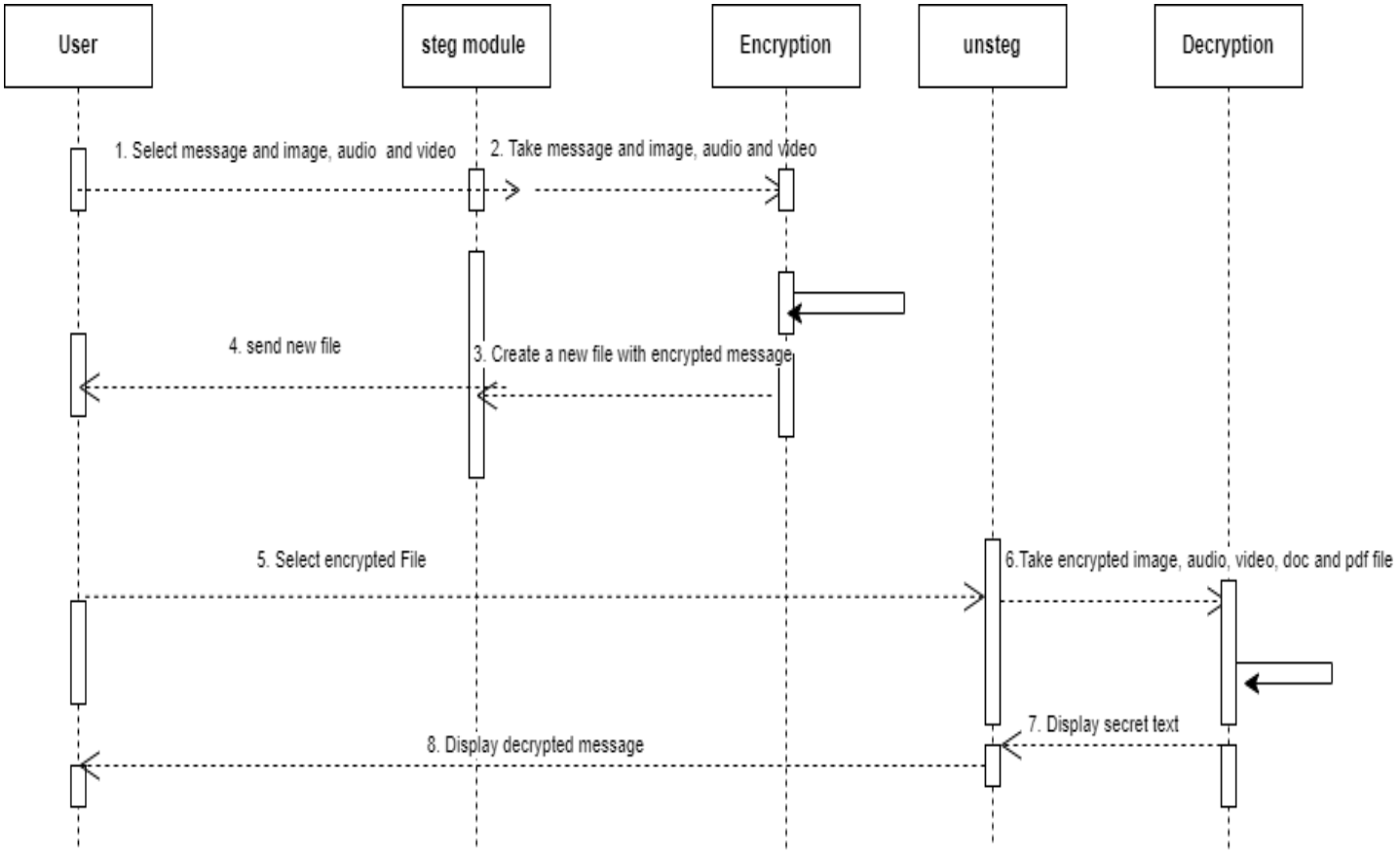
3.6.4.1. Class Diagram :



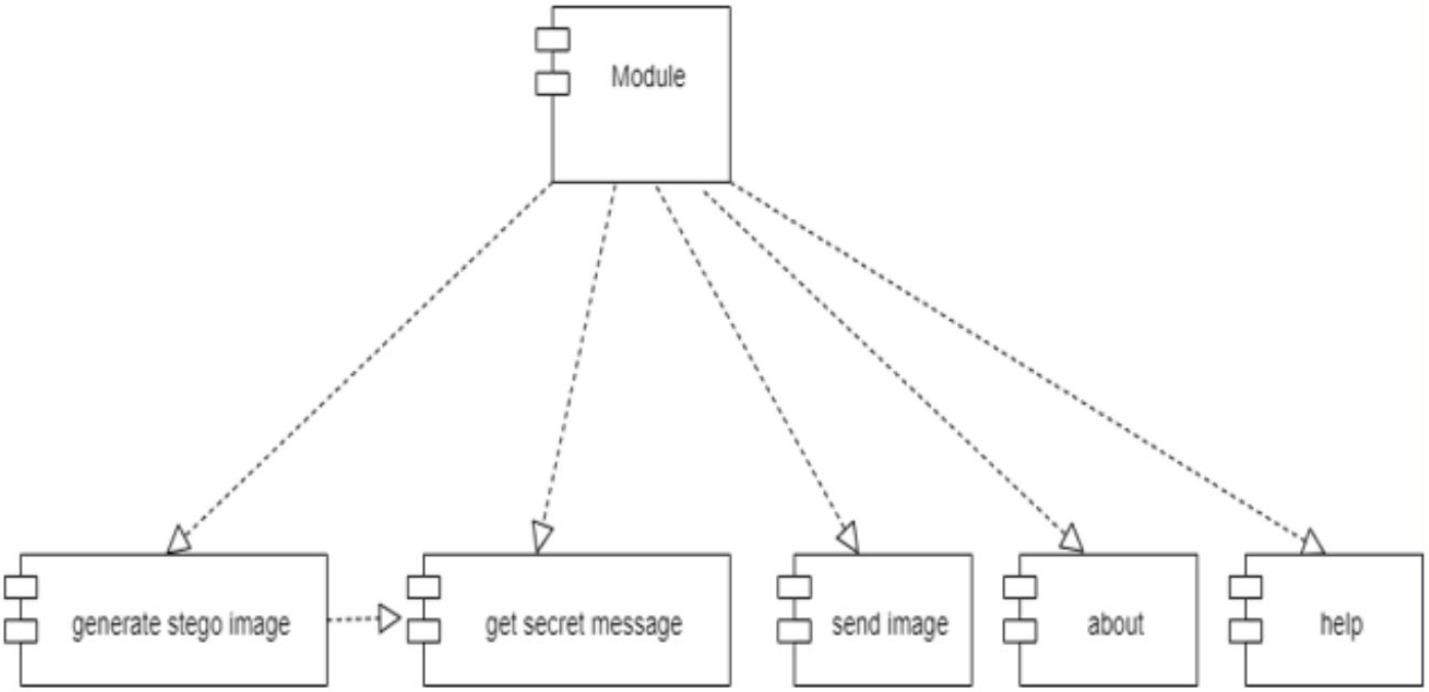
3.6.4.2. Use case Diagram :



3.6.4.3. Sequence Diagram :



3.6.4.4. Component Diagram :



3.6.5. Pseudo Codes :

3.6.5.1. Video Encryption :

```
FUNCTION encode_vid_data():
   inputttt = dialog box to select video file with valid extensions

   cap = create a video capture object for inputttt
   vidcap = create a video capture object for inputttt

   fourcc = create a four-character code of codec to be used (in this case, XVID)
   frame_width = get the width of the frames in vidcap
   frame_height = get the height of the frames in vidcap

   size = (frame_width, frame_height)
   out = create a VideoWriter object to write to inputttt with the specified fourcc codec, 25.0
fps, and frame size

   max_frame = 0
   WHILE cap is opened:
      ret, frame = read a frame from cap
      IF ret is False:
         BREAK the loop
      max_frame += 1

   cap.release()
   PRINT "Total number of Frame in selected Video :", max_frame
   input_video_parameters()
   PRINT frame_no
   n = convert frame_no to an integer

   frame_number = 0
   WHILE vidcap is opened:
      frame_number += 1
      ret, frame = read a frame from vidcap
      IF ret is False:
         BREAK the loop
      IF frame_number equals n:
         change_frame_with = embed(frame)
         frame_ = change_frame_with
         frame = change_frame_with
      out.write(frame)

   PRINT "Encoded the data successfully in the video file."
  RETURN frame_
```

3.6.5.2. Video Decryption :

Function decode_vid_data(frame_):
Set aaa to input_video_parameters
Create a video capture object cap using cv2.VideoCapture(inputttt)
Set max_frame to 0
While cap is opened:
Read the next frame from cap and store the result in ret and frame
If ret is False, break out of the loop
Increment max_frame by 1
Print "Total number of Frame in selected Video :", max_frame
Print "Enter the secret frame number from where you want to extract data"
Set n to int(frame_no_2)
Create a new video capture object vidcap using cv2.VideoCapture(inputttt)
Set frame_number to 0
While vidcap is opened:
Read the next frame from vidcap and store the result in ret and frame
If ret is False, break out of the loop
Increment frame_number by 1
If frame_number equals n:
Call the function extract with the argument frame_
Return from the function decode_vid_data

3.6.5.3. Audio Encryption :

Import the wave module
Open the audio file in read mode
Get the number of frames and read all frames
Convert the frames to a list and then to a bytearray
Get the text to be encoded
Convert the text to binary using UTF-8 encoding
Append "*^*^*" to the end of the binary data
Convert each character of the binary data to a list of 8 bits
Loop through each bit in the binary data list
Get the binary representation of the current audio frame
If the fourth last bit of the audio frame is the same as the current bit of the binary data, leave the audio frame
unchanged
Otherwise, set the fourth last bit to 0 and the third last bit to the current bit of the binary data
Move to the next audio frame
Save the modified audio frames to a new audio file
Close the original audio file
Display a window to get the name of the output audio file
When the user enters the name of the output file, save the modified audio frames to that file
Display a message to indicate that encoding was successful

3.6.5.4. Audio Decryption :

decode_aud_data()` function:
Import the wave module
Open the audio file in read mode
Get the number of frames and read all frames
Convert the frames to a list and then to a bytearray
Initialize an empty string for the extracted binary data
Loop through each audio frame
    Get the binary representation of the current audio frame
    If the second last bit of the audio frame is 0, add the fourth last bit to the extracted binary data
    Otherwise, add the third last bit to the extracted binary data
    If the extracted binary data ends with "*^*^*", stop looping through the audio frames
Convert the extracted binary data to text using UTF-8 encoding
Display the decoded text in a window

3.6.5.5. Image Encryption :

```
BEGIN
data = input("\nEnter the data to be Encoded in Image :")
IF len(data) == 0
    THEN RAISE ValueError('Data entered to be encoded is empty')
END IF
nameoffile = input("\nEnter the name of the New Image (Stego Image) after Encoding(with extension):")
no_of_bytes = (img.shape[0] * img.shape[1] * 3) // 8
PRINT "\t\nMaximum bytes to encode in Image :", no_of_bytes
IF len(data) > no_of_bytes THEN
   RAISE ValueError("Insufficient bytes Error, Need Bigger Image or give Less Data !!")
END IF
data += '*^*^*'
binary_data = msgtobinary(data)
PRINT "\n"
PRINT binary_data
length_data = len(binary_data)
PRINT "\nThe Length of Binary data", length_data
index_data = 0
FOR i IN img DO
   FOR pixel IN i DO
      r, g, b = msgtobinary(pixel)
      IF index_data < length_data THEN
         pixel[0] = int(r[:-1] + binary_data[index_data], 2)
         index_data += 1
```

```
        END IF
        IF index_data < length_data THEN
            pixel[1] = int(g[:-1] + binary_data[index_data], 2)
            index_data += 1
        END IF
        IF index_data < length_data THEN
            pixel[2] = int(b[:-1] + binary_data[index_data], 2)
            index_data += 1
        END IF
        IF index_data >= length_data THEN
            BREAK
        END IF
    END FOR
END FOR
cv2.imwrite(nameoffile,img)
```

3.6.5.6. Image Decryption :

```
FUNCTION decode_img_data(img)
    SET data_binary to an empty string
    FOR EACH i IN img
        FOR EACH pixel IN i
            SET r, g, b to the result of calling msgtobinary(pixel)
            APPEND the last character of r to data_binary
            APPEND the last character of g to data_binary
            APPEND the last character of b to data_binary
            SET total_bytes to a list of strings where each string is a sequence of 8 characters in data_binary
            SET decoded_data to an empty string
            FOR EACH byte IN total_bytes
                APPEND the character represented by the binary value of byte to decoded_data
                IF the last 5 characters of decoded_data are "*^*^*"
                    PRINT "The Encoded data which was hidden in the Image was :--  " concatenated with decoded_data without the last 5 characters
                    RETURN
            END FOR
        END FOR
    END FOR
END FUNCTION
```

3.6.6. Decision Tables :

Inputs	Actions
Encryption key	1. Convert plaintext into binary form
Plaintext	2. Embed binary data into carrier
Carrier	3. Save carrier with hidden data
Decryption key	4. Retrieve carrier with hidden data
Carrier with data	5. Extract hidden data from carrier
Hidden data binary	6. Decrypt binary data using decryption key
Decrypted data	7. Convert binary data to plaintext
Steganographic algorithm	2.1 Apply steganographic algorithm to embed binary data into carrier
Hidden data binary	6. Decrypt binary data using decryption key
Decrypted data	7. Convert binary data to plaintext
Steganographic algorithm	2.1 Apply steganographic algorithm to embed binary data into carrier
	5.1 Apply steganographic algorithm to extract hidden data from carrier
Binary carrier	2.2 Create binary version of carrier
Binary hidden data	5.2 Extract binary hidden data from carrier using steganographic algorithm
Binary plaintext	7.1 Convert binary decrypted data to plaintext using ASCII or Unicode encoding
Error message	1.1 If encryption key is invalid or missing, return error message
	2.1 If plaintext is invalid or missing, return error message

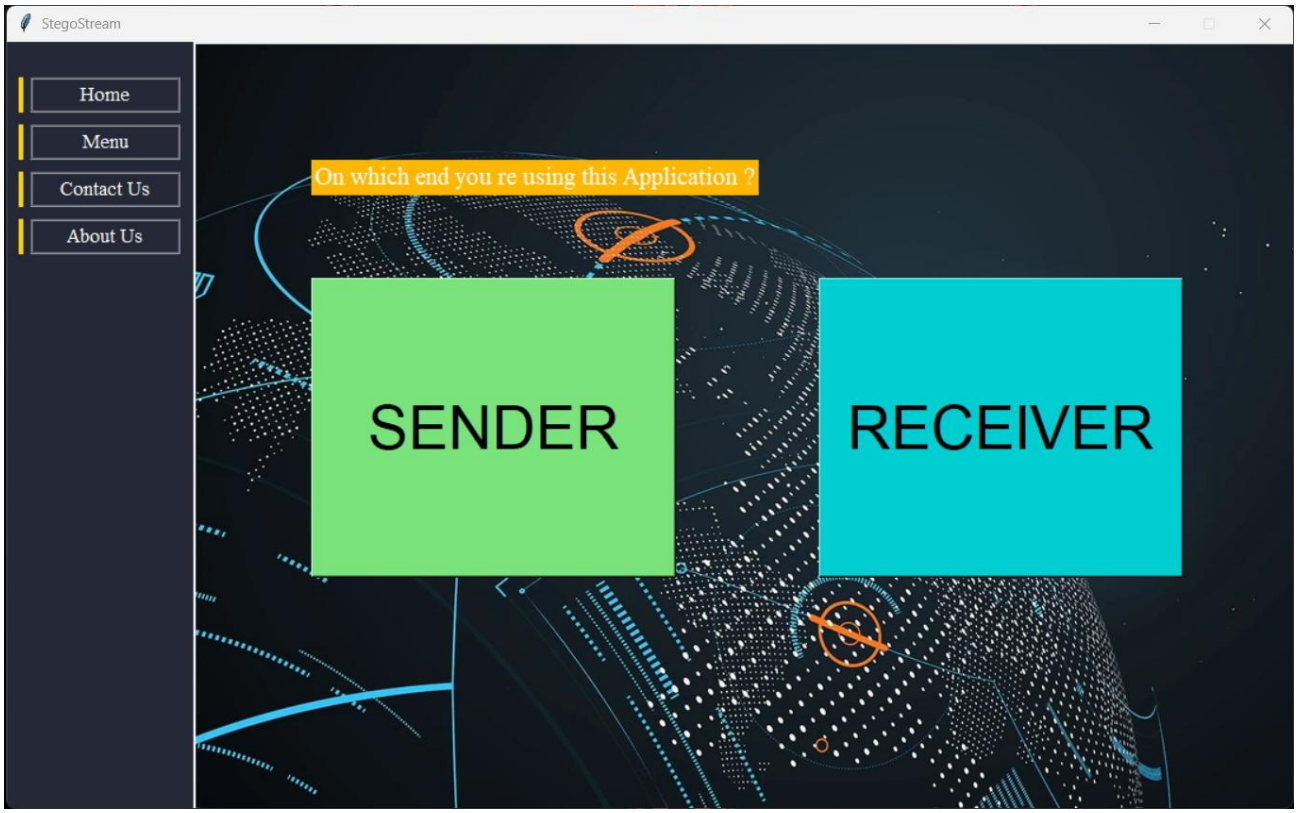
Inputs	Actions
	2.2 If carrier is invalid or missing, return error message
	4.1 If decryption key is invalid or missing, return error message
	5.1 If carrier with hidden data is invalid or missing, return error message
	5.2 If binary hidden data extraction fails, return error message
	6.1 If binary data decryption fails, return error message

3.6.7. Decision Tree :

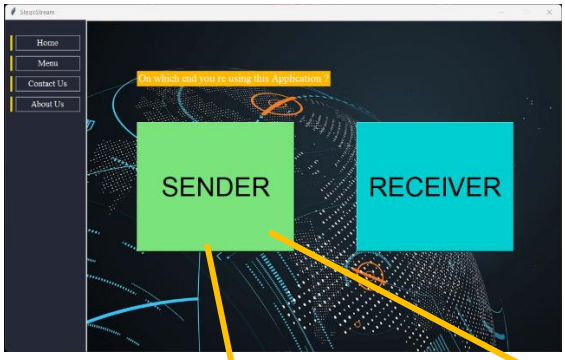
4. System Design :

4.1. Output Screens :

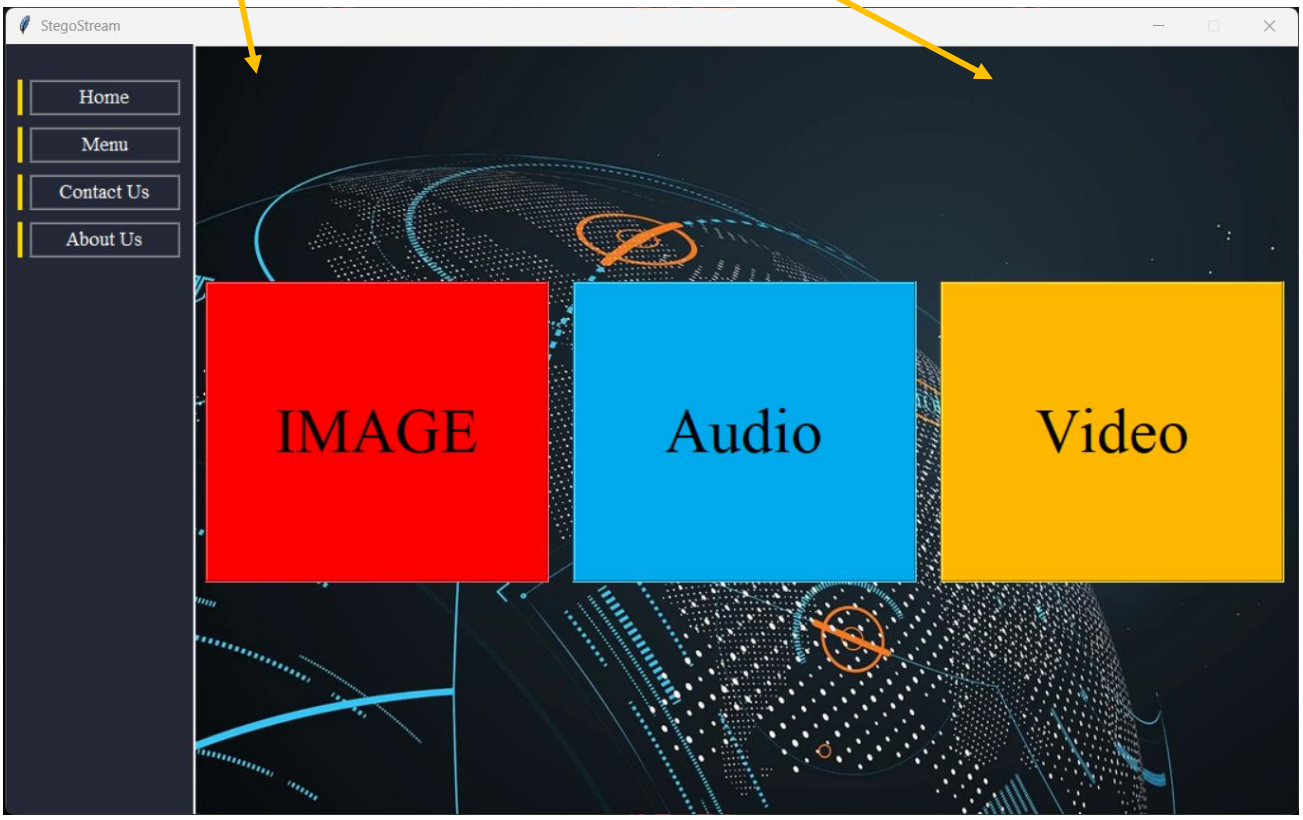
1]The Main window screen :



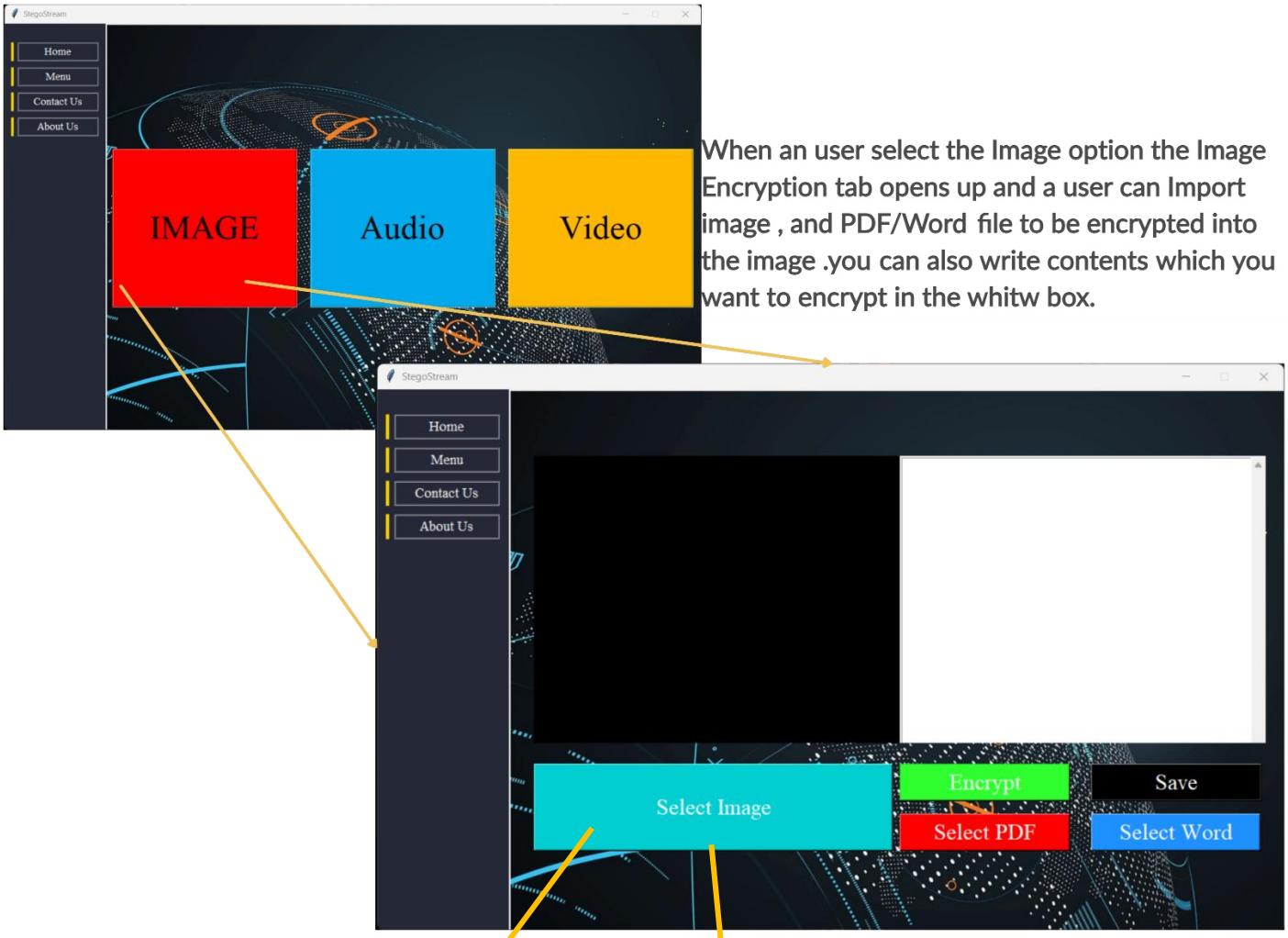
2] The SENDER functionality :



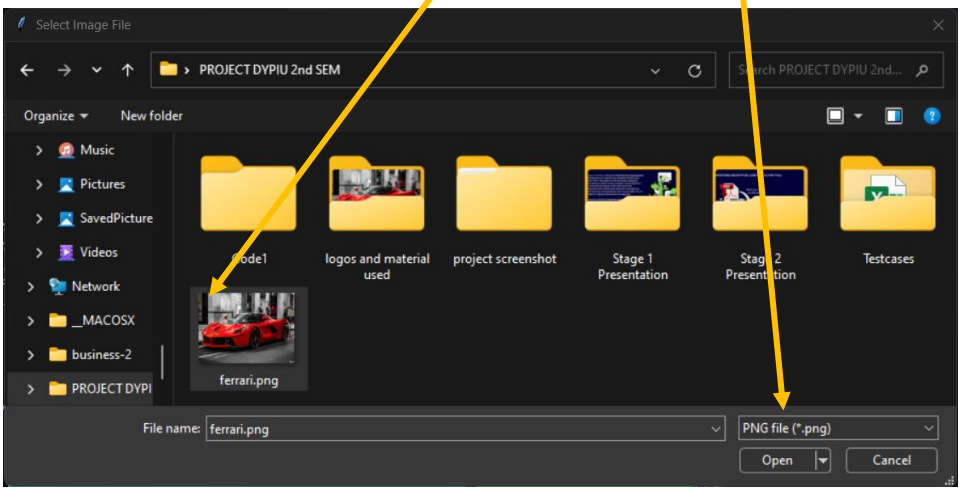
When a user click on the sender button this three options POPs Up (these options are related to encryption process as we kept it belongs to the need of sender)



3] Image Encryption :

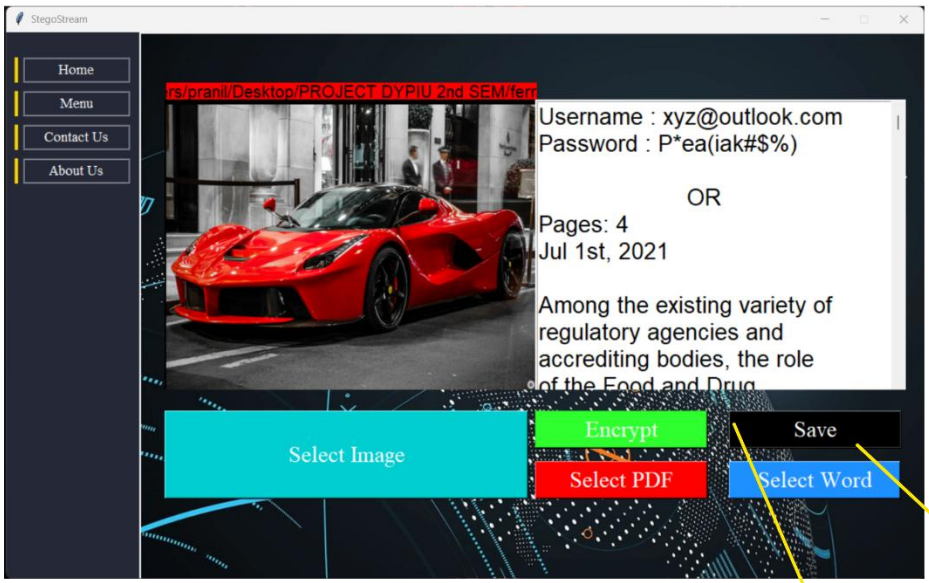


When an user select the Image option the Image Encryption tab opens up and a user can Import image , and PDF/Word file to be encrypted into the image .you can also write contents which you want to encrypt in the whitw box.

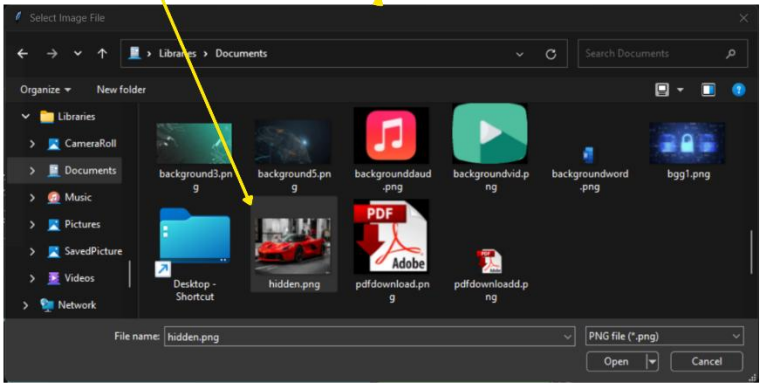


When a user clicks on Select image Button a File explorer window POPs Up and then the user is able to select Image from it(the image in which user wants to hide data).

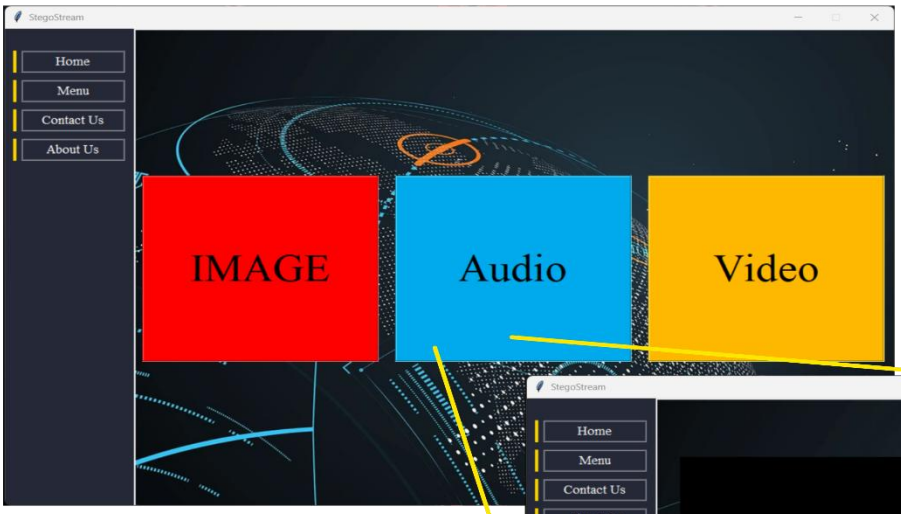
The Image that user have selected is displayed in the black container and user can input their own data in the white box and can also select PDF/Word file to encrypt in image .



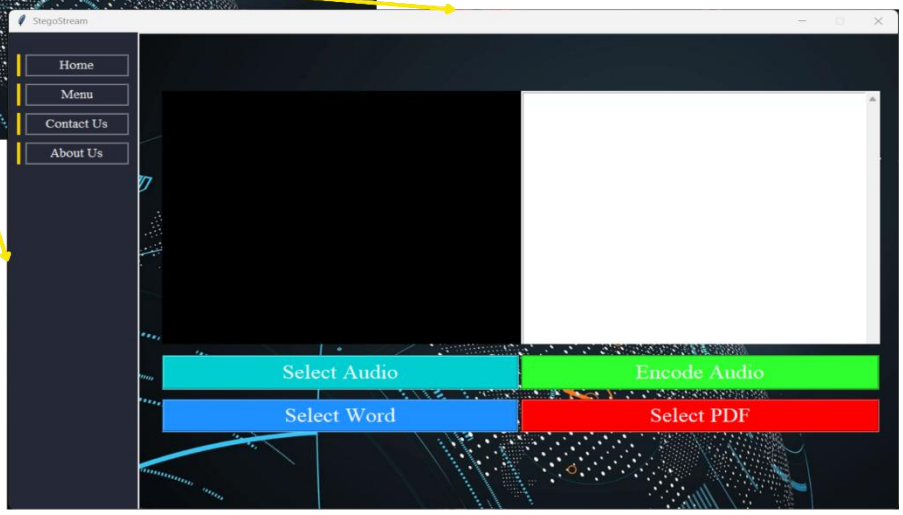
After the user clicks on encrypt button data is encrypted in this image and ,they can save this stego image by clicking on “Save” button.It will be automatically saved in documents folder.



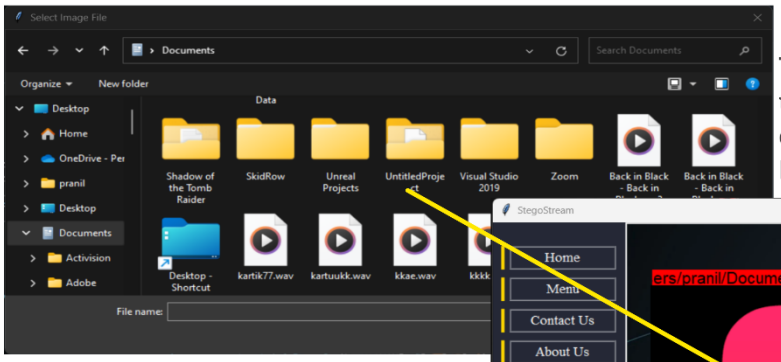
4] Audio Encryption :



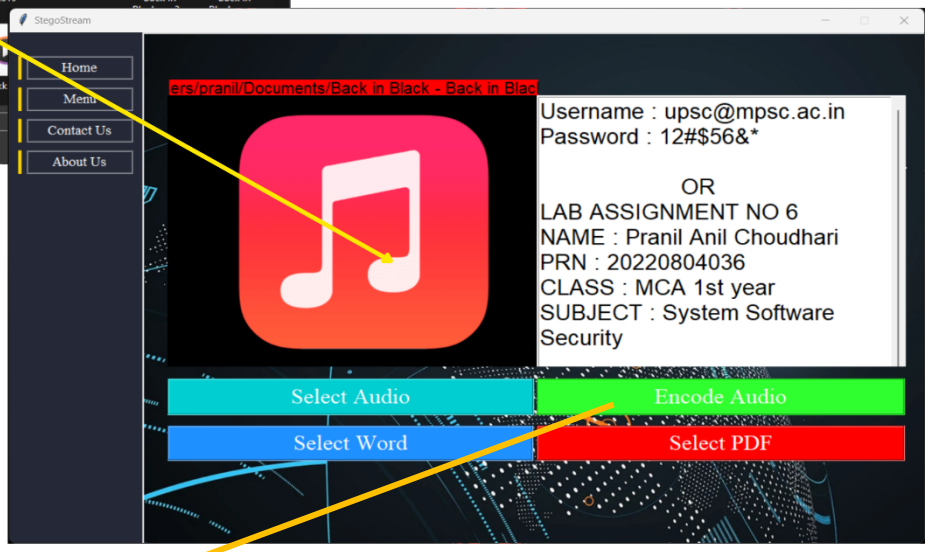
When you click on the Audio option the Audio Encryption Tab is opened.

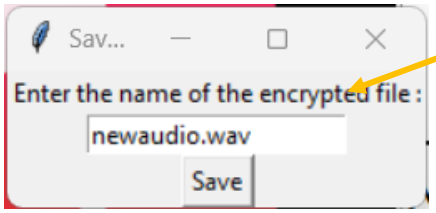


When you click on select audio,this window POPs Up .You select the desired audio in which you want to embed the secret data.



The Audio that user have selected is displayed in the black container and user can input their own data in the white box and can also select PDF/Word file to encrypt in Audio .



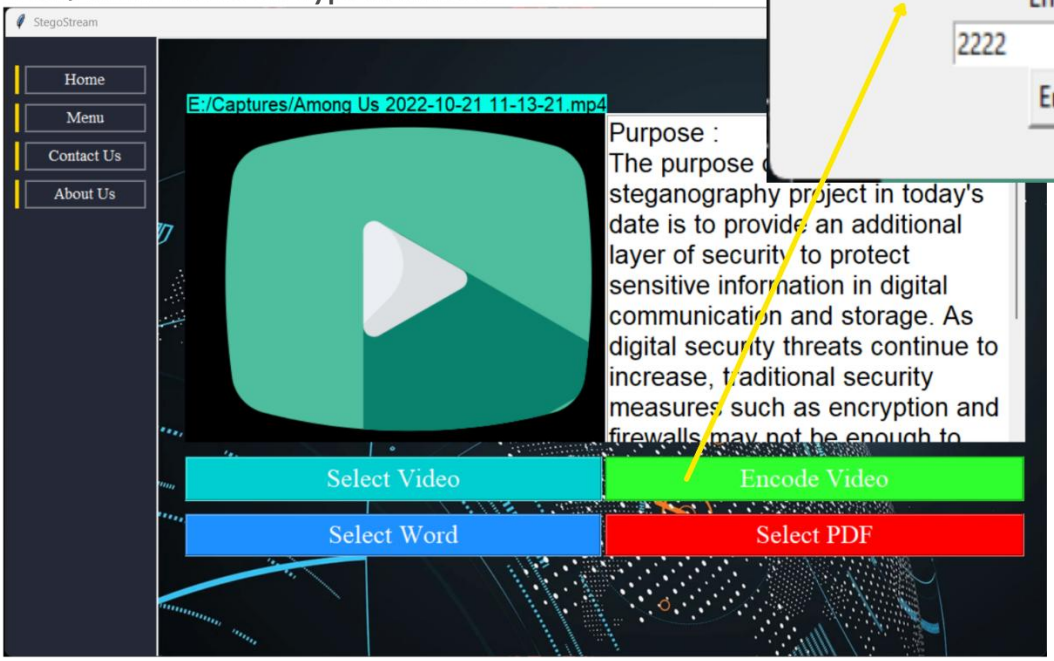


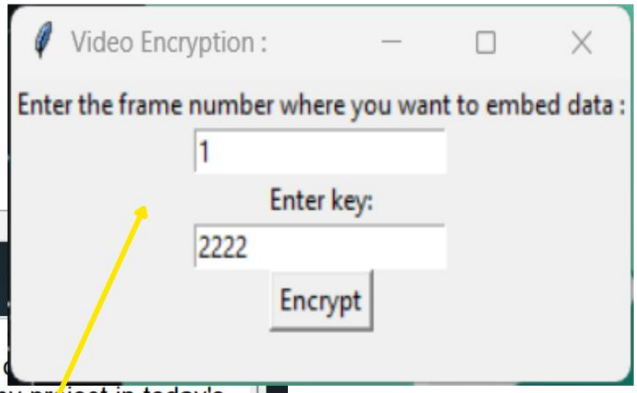
when you click on “Encode Audio” this window pops up.

5] Video Encryption :

When you click on video video encryption tab is opened

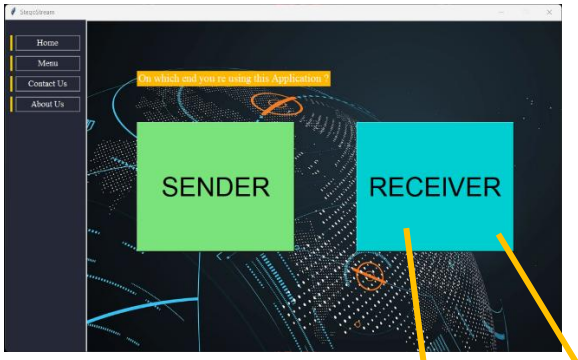
The Video that user have selected is displayed in the black container and user can input their own data in the white box and can also select PDF/Word file to encrypt in video .



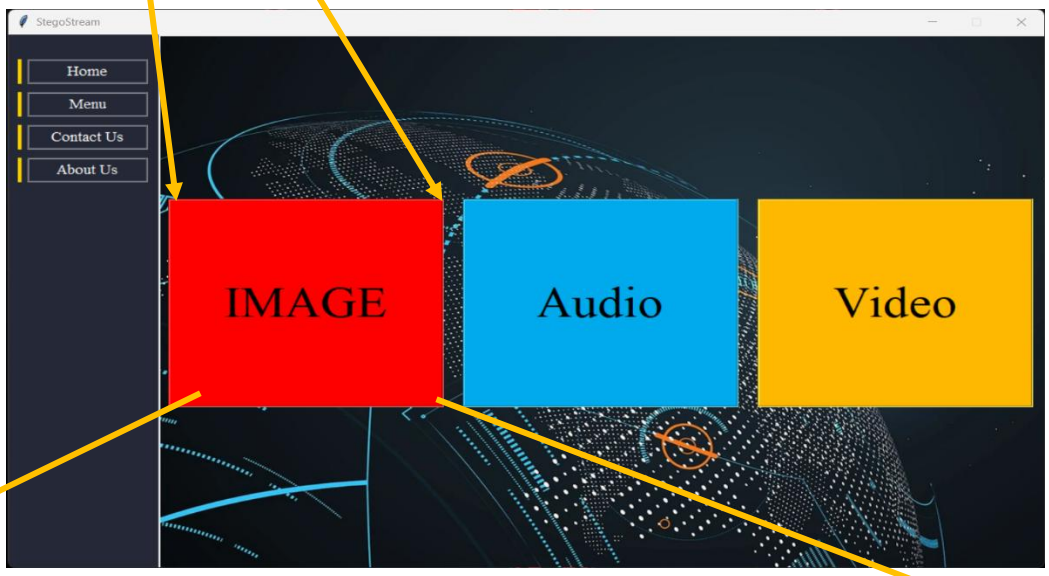


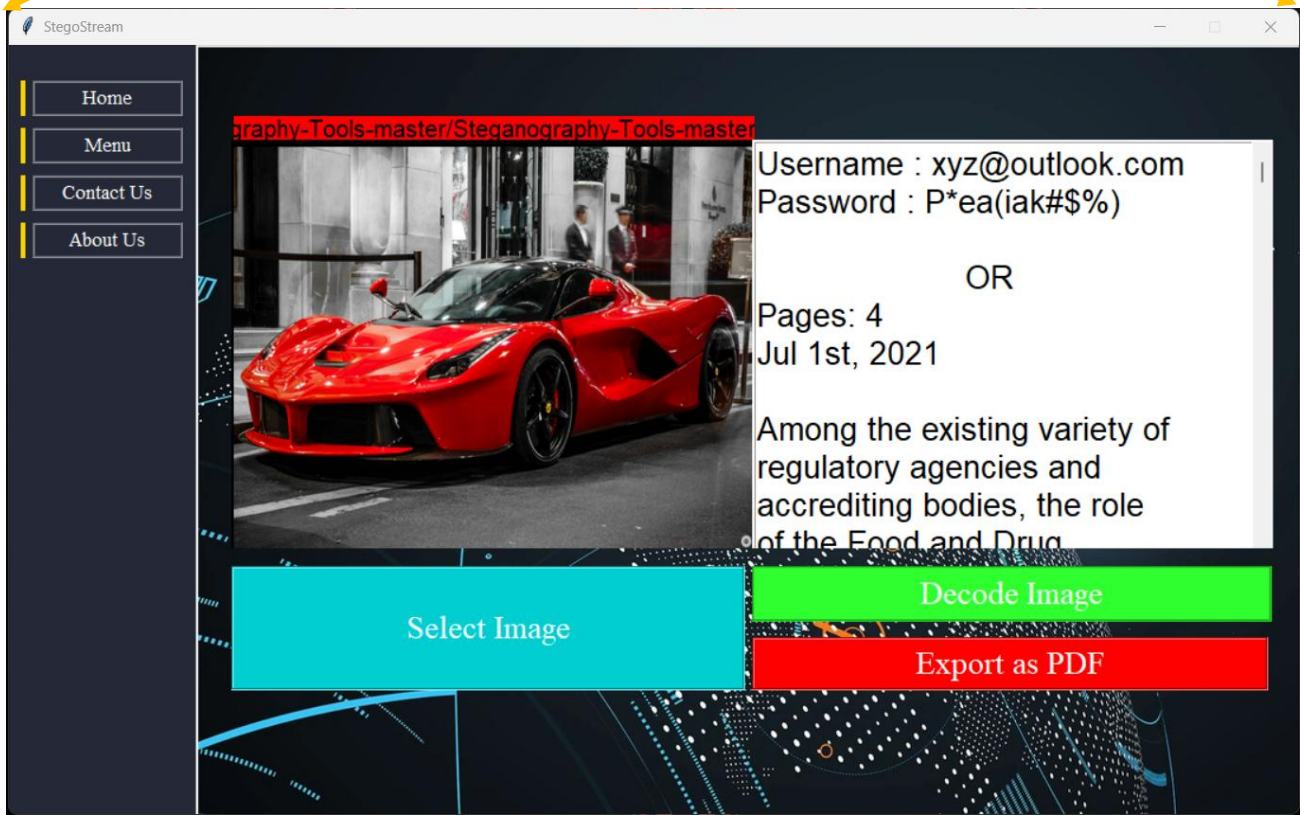
When you click on “Encode Video” this window pops up which takes frame no in which the data should be encoded and key to make it more secure

6] The RECEIVER functionality and Image Decryption:



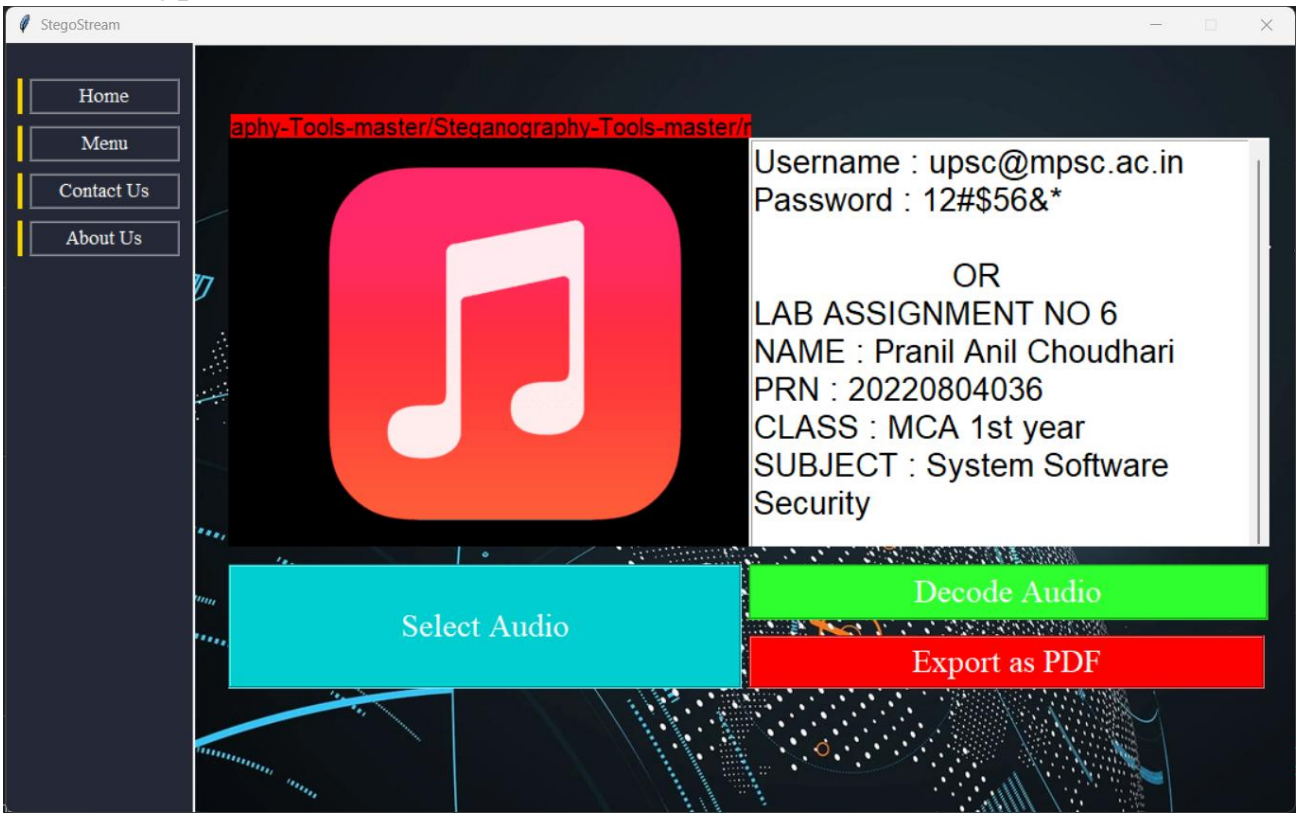
When a user click on the receiver button this three options POPs Up (these options are related to decryption process as we kept it belongs to the need of receiver)





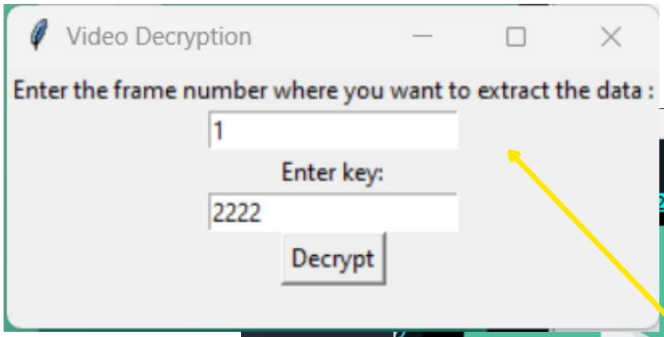
Select the hidden.png/received image and then click on “Decode Image” and you can also extract this decoded message in the form of PDF

7] Audio Decryption :

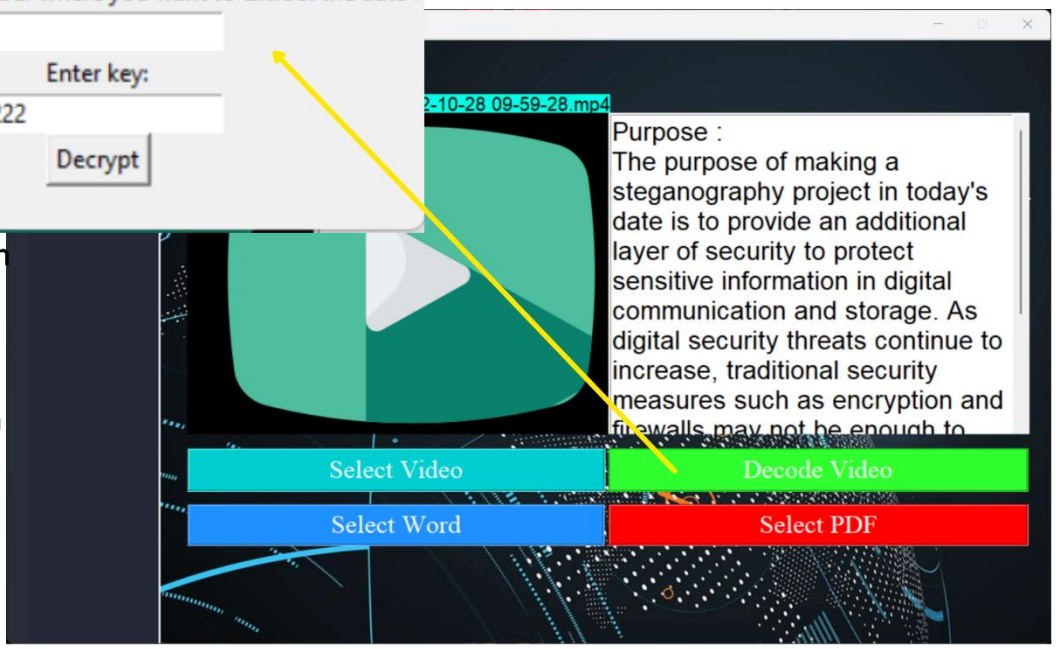


Select the received Audio file and then click on “Decode Audio” and you can also extract this decoded message in the form of PDF

8] Video Decryption :

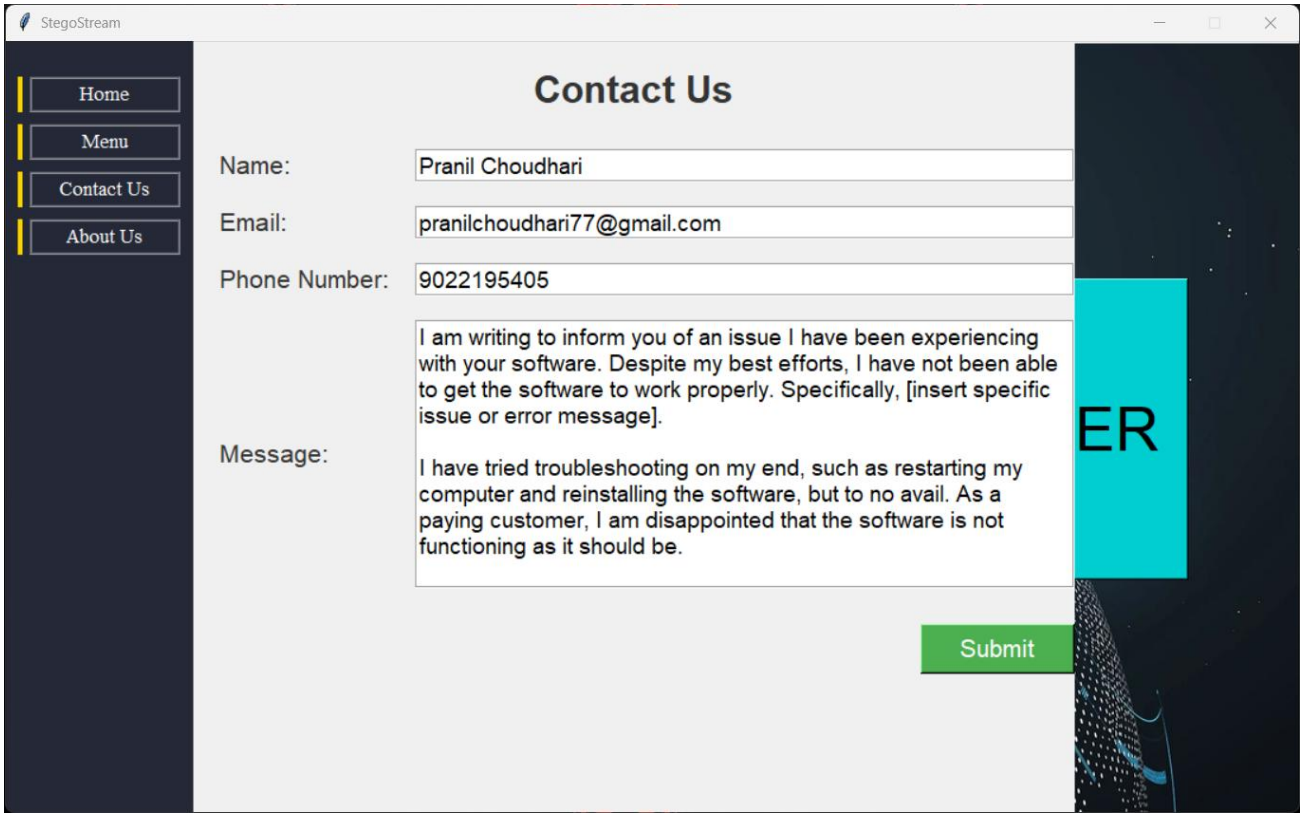


When you click on “Decode Video” this window pops up which takes frame no in which the data is encoded and key to make it more secure for decryption



Select the received Audio file and then click on “Decode Audio” and you can also extract this decoded message in the form of PDF

9] Contact us :



10] About us :

5. IMPLEMENTATION AND TESTING :

5.1. Implementation Approaches:

Image Steganography:

a. Choose a carrier image file: Select an image file in which you want to hide your secret message. You can use different file formats like PNG, JPEG, etc.
b. Create a key: Before hiding the message, create a secret key that is unique and shared only with the intended recipient.
c. Encrypt the message: Encrypt your secret message with a secure encryption algorithm such as AES, DES, etc.
d. Hide the message: Use LSB (Least Significant Bit) method to hide the encrypted message inside the carrier image. You can also use other techniques such as pixel-value differencing, etc.
e. Test the message: Verify that the message can be successfully extracted by the recipient without any errors.

Audio Steganography:

a. Choose a carrier audio file: Select an audio file such as WAV, MP3, etc. that you want to use as a carrier file.
b. Create a key: Create a secret key that is known only to you and the intended recipient.
c. Encrypt the message: Encrypt the secret message using a secure encryption algorithm such as AES, etc.
d. Hide the message: Use techniques such as LSB, phase encoding, spread spectrum, etc. to hide the encrypted message within the audio file.
e. Test the message: Verify that the message can be successfully extracted by the recipient without any errors.

Video Steganography:

- Choose a carrier video file: Select a video file such as MP4, AVI, etc. to use as a carrier file.
- Create a key: Create a secret key that is known only to you and the intended recipient.
- Encrypt the message: Encrypt your secret message using a secure encryption algorithm such as AES, etc.
- Hide the message: Use techniques such as LSB, motion vector modification, etc. to hide the encrypted message within the video file.
- Test the message: Verify that the message can be successfully extracted by the recipient without any errors.

5.2. Testing Approach :

5.2.1. Unit Testing : (Test Cases) :

Scenario TID	Scenario Desciption	Test Case ID	Pre Condition	Steps to Execute	Test Data:	Expected Result	Actual Result	Status	Post conditions
SID_ENC_VS_001	To verify the acceptance functionality of encode credentials input window for video steganography	TC_ENC_ACC_CRE_VS_001	1.Video with appropriate format must be selected. 2."Encode Video" Option should be selected/chosen 3. The selected video should have frames more than 1 4.The key should be in interger format	1.Enter the valid frame number 2.Enter the valid key 3.Click on Encrypt Button	Frame no where you want to encrypt the data : 2 Enter Key : 2222	All the values are accepted .	All the values are accepted.	PASS	The entered data should be captured by the video encryption function
SID_ENC_VS_001	To verify the acceptance functionality of encode credentials input window for video steganography	TC_ENC_ACC_CRE_VS_002	1.Video with appropriate format must be selected. 2."Encode Video" Option should be selected/chosen 3. The selected video should have frames more than 1 4.The key should be in interger format	1.Enter the invalid frame number 2.Enter the invalid key 3.Click on Encrypt Button	Frame no where you want to encrypt the data : 0 Enter Key : AB12	A popup message box to show an error "invalid frame no and key"	A popup message box to show an error "invalid frame no and key"	PASS	A Error message of invalid frame No and invalid key is displayed
SID_ENC_VS_001	To verify the acceptance functionality of encode credentials input window for video steganography	TC_ENC_ACC_CRE_VS_003	1.Video with appropriate format must be selected. 2."Encode Video" Option should be selected/chosen 3. The selected video should have frames more than 1 4.The key should be in interger format	1.Enter the invalid frame number 2.Enter the valid key 3.Click on Encrypt Button	Frame no where you want to encrypt the data : 0 Enter Key : 2222	A popup message box to show an error "invalid frame no"	A popup message box to show an error "invalid frame no "	PASS	A Error message of invalid frame No is displayed
SID_ENC_VS_001	To verify the acceptance functionality of encode credentials input window for video steganography	TC_ENC_ACC_CRE_VS_004	1.Video with appropriate format must be selected. 2."Encode Video" Option should be selected/chosen 3. The selected video should have frames more than 1 4.The key should be in interger format	1.Enter the valid frame number 2.Enter the invalid key 3.Click on Encrypt Button	Frame no where you want to encrypt the data : 2 Enter Key : ABCD123	A popup message box to show an error "invalid key"	A popup message box to show an error "invalid key"	PASS	A Error message of invalid key is displayed

Scenario TID	Scenario Desciption	Test Case ID	Pre Condition	Steps to Execute	Test Data:	Expected Result	Actual Result	Status	Post conditions
SID_DEc_VS_002	To verify the acceptance functionality of decode credentials input window for video steganography	TC_DEC_ACC_CRE_VS_001	1.Video with appropriate format must be selected. 2."Decode Video" Option should be selected/chosen 3. The selected video should have frames more than 1 4.The key should be in interger format	1.Enter the valid frame number 2.Enter the valid key 3.Click on Decrypt Button	Frame no where you want to encrypt the data : 2 Enter Key : 1234	All the values are accepted	All the values are accepted.	PASS	The entered data should be captured by the video encryption function
SID_DEC_VS_002	To verify the acceptance functionality of encode credentials input window for video steganography	TC_DEC_ACC_CRE_VS_002	1.Video with appropriate format must be selected. 2."Decode Video" Option should be selected/chosen 3. The selected video should have frames more than 1 4.The key should be in interger format	1.Enter the invalid frame number 2.Enter the invalid key 3.Click on Encrypt Button	Frame no where you want to encrypt the data : 0 Enter Key : AB12	A popup message box to show an error "invalid frame no and key"	A popup message box to show an error "invalid frame no and key"	PASS	A Error message of invalid frame No and invalid key is displayed
SID_DEC_VS_002	To verify the acceptance functionality of encode credentials input window for video steganography	TC_DEC_ACC_CRE_VS_003	1.Video with appropriate format must be selected. 2."Decode Video" Option should be selected/chosen 3. The selected video should have frames more than 1 4.The key should be in interger format	1.Enter the invalid frame number 2.Enter the valid key 3.Click on Encrypt Button	Frame no where you want to encrypt the data : 0 Enter Key : 2222	A popup message box to show an error "invalid frame no"	A popup message box to show an error "invalid frame no "	PASS	A Error message of invalid frame No is displayed
SID_DEC_VS_002	To verify the acceptance functionality of encode credentials input window for video steganography	TC_DEC_ACC_CRE_VS_004	1.Video with appropriate format must be selected. 2."Decode Video" Option should be selected/chosen 3. The selected video should have frames more than 1 4.The key should be in interger format	1.Enter the valid frame number 2.Enter the invalid key 3.Click on Encrypt Button	Frame no where you want to encrypt the data : 2 Enter Key : ABCD123	A popup message box to show an error "invalid key"	A popup message box to show an error "invalid key"	PASS	A Error message of invalid key is displayed

Scenario TID	Scenario Desciption	Test Case ID	Pre Condition	Steps to Execute	Test Data:	Expected Result	Actual Result	Status	Post conditions
SID_VS_001	To verify the functionality of video steganography.To check if the contents gets encoded into the provided video	TC_VS_EN_1	1.Video with appropriate format must be selected. 2."Encode Video" Option should be selected/chosen 3.The credentials required for encoding such as frame number and key should be entered correctly.	1.Select a video 2.Enter the text data or select the PDF to be encoded into the video 3.Selct the encode option 4.Enter the frame no where you want to save data . 5.Enter the encryption key which you will share with the receiver	1. Video 2.Data which you want to encrypt 3.frame number 4.Encryption key	The Data is Successfully encrypted in the video	The Data is Successfully encrypted in the video	PASS	The Command is again returned to the GUI .
SID_VS_002	To verify the functionality of video steganography.To check if the contents are decoded from the provided video	TC_VS_DC_2	1.Video with appropriate format must be selected. 2."Encode Video" Option should be selected/chosen 3.The credentials required for encoding such as frame number and key should be entered correctly.	1.Select a video 2.Selct the Decode option 3.Enter the frame no where your data is stored. 4.Enter the encryption key which sender given to you	1. Video 3.frame number 4.Encryption key	The Data is Successfully Decrypted from the video	The Data is Successfully Decrypted from the video	PASS	The Command is again returned to the GUI .

5.2.2. Integration Testing : (Test Cases) :

Scenario TID	Scenario Desciption	Test Case ID	Pre Condition	Steps to Execute	Test Data:	Expected Result	Actual Result	Status	Post conditions
SID_GUI_SS_MP_001	To verify the acceptance functionality of GUI , to check if it can accept an Image	TC_GUI_SS_IMG_1	1.The Executable File of StegoStream application (Main file of the program) must be running .	1.click on "Select Image" option 2.when the file Explorer Pops up then select the an Image in which you want to encrypt the data	An image of your chice from your Files.	Image is displayed in the GUI.	Image is displayed in the GUI	PASS	The selected image must be displayed in the GUI
SID_GUI_SS_MP_002	To verify the acceptance functionality of GUI , to check if it can accept Video	TC_GUI_SS_VID_2	1.The Executable File of StegoStream application (Main file of the program) must be running .	1.click on "Select Video" option 2.when the file Explorer Pops up then select the a video in which you want to encrypt the data	A video of your chice from your Files.	The video thumbnail/icon and path is displayed in the GUI.	The video thumbnail/icon and path is displayed in the GUI.	PASS	The video thumbnail/icon and path of the Selected video is displayed in the GUI.
SID_GUI_SS_MP_003	To verify the acceptance functionality of GUI , to check if it can accept Audio	TC_GUI_SS_AUD_3	1.The Executable File of StegoStream application (Main file of the program) must be running .	1.click on "Select Audio" option 2.when the file Explorer Pops up then select the a Audio in which you want to encrypt the data	A Audio of your chice from your Files.	The Audio thumbnail/icon and path is displayed in the GUI.	The Audio thumbnail/icon and path is displayed in the GUI.	PASS	The audio thumbnail/icon and path of the Selected audio is displayed in the GUI.
SID_GUI_SS_MP_004	To verify the acceptance functionality of GUI , to check if it can accept PDF	TC_GUI_SS_PDF_4	1.The Executable File of StegoStream application (Main file of the program) must be running .	1.click on "Select PDF" option 2.when the file Explorer Pops up then select the a PDF in which you want to encrypt the data	A PDF of your chice from your Files.	The Contents of the PDF are displayed in the Text box of the GUI	The Contents of the PDF are displayed in the Text box of the GUI	PASS	The Contents of the PDF are displayed in the Text box of the GUI
SID_GUI_SS_MP_005	To verify the acceptance functionality of GUI , to check if it can accept Word	TC_GUI_SS_WRD_5	1.The Executable File of StegoStream application (Main file of the program) must be running .	1.click on "Select Word" option 2.when the file Explorer Pops up then select the a Word document in which you want to encrypt the data	A word file of your chice from your Files.	The Contents of the Word are displayed in the Text box of the GUI	The Contents of the Word are displayed in the Text box of the GUI	PASS	The Contents of the Word are displayed in the Text box of the GUI

5.3. Modifications and Improvements:

- Increase the embedding capacity: One possible improvement to the steganography system could be to increase the amount of data that can be hidden within the cover media. This could be achieved by improving the compression algorithms or by finding more efficient ways to store data within the cover media.

- Enhance the security: The security of the steganography system could be improved by incorporating stronger encryption algorithms to protect the embedded data. The system could also be modified to prevent detection by anti-steganography tools.

- Support for more file types: The steganography system currently supports image, audio, and video files. However, there may be other file types that could benefit from steganography. Adding support for additional file types could improve the versatility of the system.

- User interface improvements: The user interface of the steganography system could be improved to make it more intuitive and user-friendly. This could involve redesigning the interface, adding tooltips and help messages, and simplifying the workflow.

- Performance optimization: The steganography system could be optimized for performance to reduce the time it takes to embed and extract data from cover media. This could involve optimizing the algorithms used or parallelizing the processing.

- Robustness to compression: Compression of the cover media can alter the embedded data, leading to loss of information. Finding ways to make the steganography system more robust to compression could improve its effectiveness.

- Improved metadata handling: Metadata can reveal information about the embedded data, which can compromise the security of the steganography system. Finding ways to handle metadata more effectively could improve the security of the system.

- Integration with other systems: The steganography system could be integrated with other security systems to provide an additional layer of protection. For example, it could be integrated with a firewall or intrusion detection system to detect and prevent unauthorized access to the embedded data.

6. CONCLUSIONS:

6.1. Conclusion :

The steganography project successfully implemented various techniques of hiding data in digital media such as images, videos, and audio . Based on the successful implementation of various steganography techniques in the project, it can be concluded that steganography is an effective way of securing data in digital media. The project has also highlighted the importance of data security in today's digital world and the need for advanced techniques to ensure the confidentiality and integrity of sensitive information. Furthermore, the project has opened up opportunities for further research and exploration of more advanced steganography techniques such as using neural networks. This can lead to the development of more secure and robust steganography methods that can withstand attacks from sophisticated adversaries. In addition to research, the project has practical applications in digital watermarking and copyright protection. These techniques can be used to protect intellectual property rights and prevent unauthorized use or distribution of copyrighted material.

Overall, the project has been a valuable learning experience, providing hands-on experience in implementing steganography techniques and highlighting the importance of data security in today's digital landscape.

6.2. Limitations of the System :

- **Data capacity:** The amount of data that can be hidden in an image is limited by the size of the image and the steganography technique used. If you try to hide too much data, the image quality may be compromised, or the hidden data may become too obvious.

- **Security:** While steganography can be used for hiding sensitive information, it is not a substitute for encryption. If someone gains access to the image, they may be able to detect the presence of hidden data or even extract it with the right tools.

- **Detection:** It is possible to detect the presence of hidden data in an image by analyzing its statistical properties. To ensure the effectiveness of steganography, it is important to use techniques that make it difficult for an attacker to detect the presence of hidden data.

- **Robustness:** Steganography techniques can be vulnerable to attacks that attempt to modify or manipulate the image. It is important to use robust techniques that can withstand attacks and preserve the hidden data even if the image is modified.

- **Computational complexity:** Some steganography techniques can be computationally expensive, which may limit their applicability in real-time applications or on low-power devices.

- **Compatibility:** Some steganography techniques are only compatible with certain types of images or data formats. If the image or data format is not supported, the steganography technique may not work or may produce poor results.

6.3. Future Scope :

- **Implement advanced encryption techniques:** To enhance the security of your steganography project, you can implement advanced encryption techniques like Elliptic Curve Cryptography (ECC), Homomorphic Encryption (HE), Quantum Key Distribution (QKD) to encrypt the hidden data before embedding it into the cover medium.

- **Improve the payload size and capacity:** You can work on improving the payload size and capacity of your steganography project, which refers to the amount of data that can be hidden in a cover medium. You can explore various steganography techniques to increase the payload size and capacity without affecting the quality of the cover medium.

- **Enhance the robustness of the steganography system:** In some cases, the hidden data may be lost or damaged during transmission, which can compromise the security of the steganography system. To address this issue, you can implement error-correction techniques to ensure the robustness of the steganography system.

- **Develop a user-friendly interface:** A user-friendly interface can make your steganography project more accessible to users who are not familiar with steganography. You can design an intuitive and easy-to-use interface that allows users to easily embed and extract data from cover media.

- **Multi-layer steganography:** Multi-layer steganography is a technique that involves hiding data in multiple layers of a cover medium. This technique can increase the payload capacity of the steganography system and make it more resistant to detection

7. REFERENCE :

1) An Overview of Different Type of Data Hiding Scheme in Image using Steganographic Techniques MukeshGarg* A.P. GurudevJangraM.Tech. Scholar H.O.D in CSE Department Jind Institute of Engineering & Technology Jind Institute of Engineering &Technol, Volume 4, Issue 1, January 2014 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering Research Paper is Available online at: www.ijarcsse.com

2) International Refereed Journal of Engineering and Science (IRJES) ISSN (Online) 2319-183X, (Print) 2319-1821 Volume 6, Issue 1 (January 2017), PP.68-71, Survey Paper on Steganography Namrata Singh Computer Science and engineering ABES Engineering College, Ghaziabad A.K.T.U.

3) American Journal of Engineering Research (AJER) e-ISSN : 2320- 0847 p-ISSN : 2320-0936 Volume-02, Issue-11, pp-122-128 www.ajer.org Steganography: A Review of Information Security Research and Development in Muslim World YunuraAzuraYunus, SalwaAbRahman, Jamaludin Ibrahim Kuliyyah of Information and Communication Technology International Islamic University Malaysia

4) International Journal of Innovative Research in Computer and Communication Engineering Vol. 4, Issue 1, January 2016,A Survey Paper on Steganography Techniques Dr. Rajkumar L Biradar1, Ambika Umashetty2 Associate Professor, Dept. of Electronics and Telematics, G. Narayanamma Institute of Technology & Science, Hyderabad, India1 Dept. of Computer Science & Engineering, Appa Institute of Engineering & Technology, Kalaburagi.

5) T. Morkel , J.H.P. Eloff and M.S. Olivier “An Overview of Image Steganography”.

6) SamerAtawneh, Ammar Almomani1 and Putra Sumari, “Steganography in Digital Images: Common Approaches and Tools,”IETE Technical Review, Vol 30, Issue 4, Jul-Aug 2013.

7) Mastering C# (Paperback), SQL Server Bible (Paperback) ,.NET Black Book (Paperback) books