# INTRODUCTION

This Project “Data Security through Steganography and cryptography” is a desktop application that can be used to encrypt the information using provided encryption algorithms and hide the information at the plane site, using other form of media like image, audio and pdf files.

## **Project Overview**

Here ,a file containing the information is encrypted using AES-256 encryption algorithm, then the encrypted file is encoded in one of the carrier media,(a .png format image , wav format audio , mov video file ,a pdf file).Now the final product we have is a stegano-media file.

Now to retrieve the hidden information, stegano-media is first decoded to extract the cryptic information and then the cryptic information is decrypted using AES-256 decryption algorithm. Since this is a standalone desktop application the parties involve in sharing and storing the information has to agree on the key required for the encryption and decryption.

And the key required for encrypting and decrypting the information should be agreed beforehand

## **PROBLEM STATEMENT**

The search for Fast and efficient medium for storing and, transferring the information has led to the rapid development of digital media technology. Technology available to us has provided us time and space efficient way to store and transfer those data but still those information are prone to attack at host or during the transmission. Thus we required various security measure that are to be taken like cryptography, VPN tunneling, steganography, Access restriction and so on.

The art of cryptography has been used to encrypt the information in order to conceal the actual information that is present in the data file. Cryptography conceals the actual content of the information but not the existence of the information. One of the best measure that could be taken to prevent unauthorized access it to hide the information such that its existence itself is hidden.

Combining different Security measure is only the option to provide robustness in our system, therefore this system of will help to maintain the security of the information by combining the art of cryptography and steganography.

## **PROJECT OBJECTIVES**

This project is concerned with achieving the data security by hiding the existence of the data itself. Therefore, the main objective of the project are:

i) To hide the confidential information in other form of media using steganography, and adding another layer of security through cryptography technique through AES-256

ii) To extract the confidential information form the encrypted text which is embedded in stegano-media.

## **PROJECT** **SCOPE AND IMPORTANCE**

The scope of this project are:

1. To remove the curiosity of the attacker or eavesdropper from actual sensitive data in case intrusion in the system or the network

2. It encrypt file, for the purpose of masking the actual information.

3. It decrypt file, to unmask the actual information that was altered using encryption algorithm.

## **PROJECT LIMITATIONS**

The Limitation of the project are:

1. Only lossless image file format could be used like .png, .tiff

2. Audio as the carrier media, this project only allows .wav audio format.

3. Since this is a standalone desktop application, the exchange of the keys and algorithm to be used should be done through another secure channel.

# LITERATURE STUDY

## **Review**

The best security measure that concerned entities could take to protect their confidential data is to conceal its existence. The next step that could be taken to protect those data is to encrypt the data with a robust encryption algorithm, such that a third party which got hold of the information could not figure out the actual content of those data. Thus this project will implement the security practice of Steganography and cryptography.

Steganography is the art and science of hiding communication of the information. This project system thus embeds hidden content in unremarked- able cover media so as not to arouse an eaves-dropper’s suspicion.

Cryptography converts data into a format that is unreadable for an unauthorized user, allowing it to be transmitted without unauthorized entities decoding it back into a readable format, thus maintaining the security of the data. This Project intend to provide user with a symmetric encryption algorithm like AES-256.

## **EXISTING SYSTEM**

### **OpenegSto**

OpenStego provides data hiding as well as Watermarking.OpenStego perform Steganography effectively with image files of type JPEG, JPG, BMP, GIF, PNG etc. The output of OpenStego is a PNG file. It is an open source and free Steganography tool developed using Java. It also provide watermarking which is is used to detect an unauthorized copy of image files. But it does not support audio steganography and encryption of the files.

### **RSteg**

RSteg is also image Steganography tool developed using Java. Performing Steganography using RSteg is simple. All that is require is an Image file, text to be encrypted and password to be set for decryption. The final output is stored as PNG. The stegano-image plug into the same Steganography detection tool for decryption along with a password.

Although similar kind of software are available to perform the steganography on both images and audio steganography ,the is no any software that are equipped with all the feature like steganography using audio , image , video and pdf files with additional encryption technique. Thus this project of our tried to bring together all this features in a single desktop program.

## **BASIC STRUCTURE OF STEGANOGRAPHY**

The basic structure of Steganography is made up of three components.

1. **Carrier** - The carrier can be a painting, a digital image, audio file, even a TCP/IP packet among other things. It is the object that will ‘carry’ the hidden message.

2. **Message** - The message (hidden) is being carried by the object (carrier).

3. **Key** - A key is used to decode/decipher/discover the hidden message. This can be anything from a password, a pattern, a black-light.

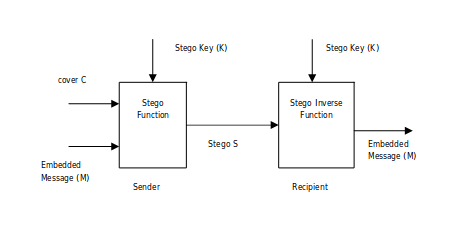


Figure 1.Basic Steganography Model

Here, the carrier media is given as input along with the file that is to be hidden in that media. The file is hidden into the carrier media using the steganographic algorithm I.e. LSB algorithm in this project. In steganalysis, the message or the hidden file is extracted from the carrier media.

## **CRYPTOGRAPHY VS STEGANOGRPAHY**

Steganography is not the same as cryptography. Cryptography hides the contents of a secret message from a malicious people, whereas steganography conceals the existence of the message. In cryptography, the structure of a message is scrambled to make it meaningless and unintelligible unless the decryption key is available. In contrast, steganography does not alter the structure of the secret message, but hides it inside a cover media so it cannot be seen.

## **STEGANOGRAPHY TECHNIQUES:**

Digital data can be embedded in many ways into the carrier media, Using either the spatial domain (LSB Replacement, Matrix embedding, Histogram modification) or transform domain (Discrete cosine Transform, Fast Fourier Transform) steganographic algorithm, the message is hidden into the cover media and the stego media is obtained. The most common techniques of data hiding in images are:

**1. Appending data bytes at the end of carrier:** The secret data bytes are appended at the end of the carrier media such as image andthe carrier media is then compressed to its original size to reduce the suspects of havingsecret data. Advantage is that it is very easy to implement. Disadvantage is it is very easy to detect and get the message.

**2. Transform domain based embedding**: Transform Embedding Techniques embed the data by modulating coefficients in a transform domain, such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) (used in JPEG compression) or Discrete Wavelet Transform (DWT). Modifying the transform coefficients provides more robustness to the compression (especially to lossey), cropping, or some image processing, than LSB techniques.

**3. Least significant bit (LSB) insertion:**

LSB techniques embed the message bits directly into the least significant bit plane of the cover image in a deterministic sequence. Here the binary representations of the secret data have been taken and the LSB of each byte is overwritten. This results in a change with too low amplitude to be human-perceptible. LSB embedding is simple, popular and many use these technique.

## **LSB Encoding Algorithm**

1. Carrier media and the Message file along with the key is inputted.
2. Convert the message file into the binary format and generate the stream of bits
3. Bytes representing the carrier-media is taken in a single array and byte stream is generated.
4. Message bits are taken sequentially and then are placed in LSB of the byte representing the carrier media
5. Repeat the step 4. Till all the message bits are placed in image.

*Output: stegno-media*

## **LSB Decoding algorithms**

1. The Stegano-media and the key is inputted.
2. Array of the bytes are generated.
3. The total number of bits of message and the bytes representing the stego-media are taken.
4. The bits stream of the message is generated.
5. Available bits are grouped to form bytes such that each byte represents single ASCII character.
6. Character are stored in the text file.

*Output: Recovered hidden message text file.*

## **Advanced Encryption standard**

The **Advanced Encryption Standard** (**AES**), also known by its original name **Rijndael** is a specification for the encryption of electronic data established by the US. National Institute of Standard and Technology in 2001. [*Wikipedia*].

AES is widely adopted symmetric encryption algorithm after its declassified .In 2003, The US government begin to use AES as the encryption standard for protecting classified information. AES is a symmetric key symmetric block cipher. It comprises three block ciphers: AES-128, AES-192, and AES-256.Each cipher encrypts and decrypts data in the block of 128bits using the cryptographic keys of 128,192, 256 bits respectively.

### **Operation of AES**

AES is an iterative cipher .It is based on the substitution-permutation network’. It comprises of a series of linked operation, some of which involve replacing inputs by specific outputs (substitution) and permutation.

AES performs all its computations on bytes rather than bits. Hence AES treats the 128 bits of a plain text block as 16bytes.These 16bytes are arranged in four columns and four rows for processing as a matrix.

The AES encryption algorithm defines a number of transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array; after which the cipher transformations are repeated over a number of encryption rounds. AES uses 10 rounds for 128 bit keys, 12 rounds for 192 bit keys and 14 rounds for 256 bit keys. Each of these rounds uses different 128-bit round key which is calculated from the original AES key.

*{picture}*

#### **Encryption process**

The AES encryption algorithm defines a number of transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array; after which the cipher transformations are repeated over a number of encryption rounds. AES uses 10 rounds for 128 bit keys, 12 rounds for 192 bit keys and 14 rounds for 256 bit keys. Each of these rounds uses different 128-bit round key which is calculated from the original AES key.

{picture}

AES consists of three different types of layers, each of them manipulates all 128 bits of the data path (also called states).

Each round, with the exception of the first, consists of all three layers and they are

1. Bytes Substution layer:

2.Diffusion layer:

3. Add round key:

#### **Decryption Process**

The Process of decryption of an AES cipher text is siilar to the encryption process in the reverse order.Each round consists of the four processes conducted in the reverse order

add round key

mix columns

shift rows

byte substitution

## **Block cipher and modes of operation**

A block cipher processes the data block of fixed size. Usually the size of a message is larger than the block size. Hence, the long message is divided into a series of sequential message blocks, and the cipher operates on these blocks one at a time. The strength of the encryption scheme depend upon the key length not the block size.

### **Modes of operation**

There are different modes of the operation of a block cipher. The different modes result in the different properties being achieved which add to the security of the underlying block cipher. Some of them are :

EBC

CBC

CFC

OFB

CTC

### **Cipher block chaining mode**

CBC mode of operation provides message dependence for generating cipher text and makes the system non-deterministic. In CBC mode, the current plaintext block is added to the previous cipher text block, the result is encrypted with the key. Decryption is thus the reverse process, which involves decrypting the current cipher text and then adding the previous cipher text block to the result.

{figure}

# METHODOLOGY

## **Method**

For the development of this project we choose to use Incremental Model as this model provide an easy and effective way to work on one function at a time and gradually completing the all other function to give final fully- functional software.

## **Model**

### **Incremental Model**

The framework we will be using for developing this project is Incremental Model. This model combines linear sequential model with the iterative prototype model. New functionalities will be added as each increment is developed. The phases of linear sequential model are: Analysis, Design, Coding and Testing. The software repeatedly passes through these phase in iteration and an increment is delivered with progressive changes. Total of 6 increment is expected to be required to complete the project.

* During 1st increment we worked on the feature that hide the data into the image file (.png).
* During the 2nd increment we worked on the feature that hide that data into the audio file (.wav format).
* During the 3rd increment we worked on the feature responsible for encrypting the file with AES-algorithm.
* During the 4th increment we worked on the feature that will hide the data into and video file.
* During 5th increment we will work on the feature responsible for hiding the text file into the pdf file
* During 6th increment we will develop the Graphical User Interface and integrate above features to build the single program

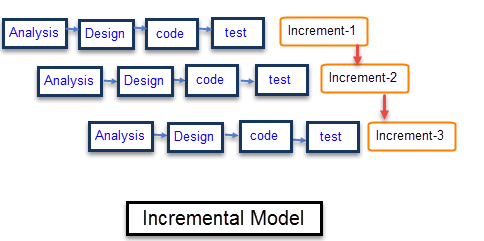


Figure 2.Incremental Model



#### **Analysis Phase**

In this phase, requirement analysis was performed in order to find out the requirements of the system.The outcome of this phase is a SRS which is an acronym for “System Requirement Specifications”.

#### **Design phase**

In this phase, the System Requirement Specification is translated into the system’s design.Here, Entity Relationship Diagram, Use Case Diagram, Flowcharts was developed.

#### **Coding Phase**

During this phase we implement our design using the python code.The programs are coded and each feature run as a indivudual mode. Intergration of these module is left to be done.

#### **Testing Phase**

During this phase Unit test on each of the four module has been successfully performed.

## **Requirement Analysis**

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting [requirements](https://en.wikipedia.org/wiki/Requirement) of the various [stakeholders](https://en.wikipedia.org/wiki/Stakeholder_(corporate)), analyzing, documenting, validating and managing software or system requirements [wiki].Analysis phase emphasizes an investigation of the problem and the requirement rather than a solution.

### **System Requirement**

*OS* compatibility

* Unix system
* Microsoft windows XP/VISTA/7/8/10.

### **Input Requirement**

The input requirement of this project are:-

* A carrier media
* A text file that has a message
* A passphrase to generate the key

### **Output Requirement**

Since the output of the program yield the steganomedia .output requirement for the program is directory where we can create and write a file.

### **Functional Requirement**

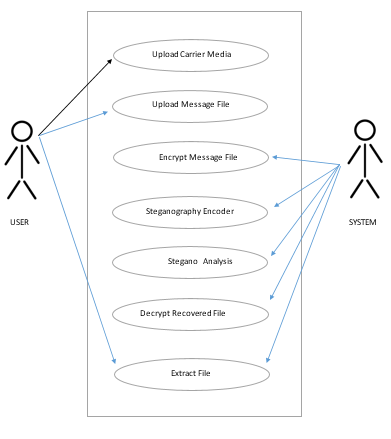
* Encrypt the message file using the encryption algorithm
* Decrypt the encrypted file using the decryption algorithm.
* Embed the encrypted file into the carrier media
* Decode the embedded encrypted file into the plain text file.

### **Interface Requirement**

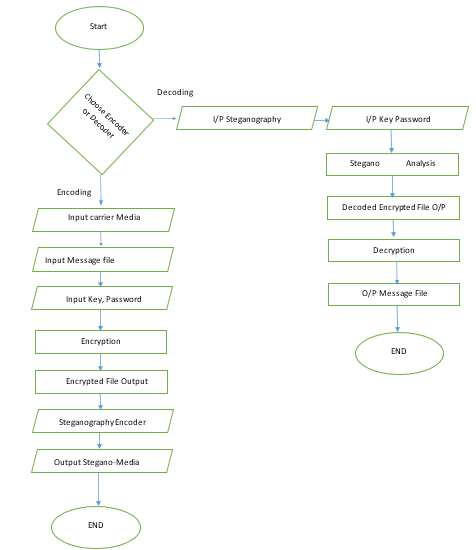
* Provide user to select the image and message file from the memory
* Provide user the interface to input the password and the key

## **System Design**

### **Use case:**

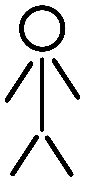


### **Flowchart**



### **System Sequence Diagram**

SYSTEM



Display Message

Output recovered text

Decryption

Decoding

Decoder

Input password

Input steganomedia

Display Menu for Input

Choose Function

Display Message

O/P Steganography Media

Encryption

Encoding

Input password

Encoder

Display Menu

Input message file

Input carrier media

Display menu for Input

Choose Function

GUI

## **Tasked Done So Far**

1. Program to encrypt and decrypt the text file.
2. Program to hide the text file into the Image
3. Program to hide the text file into the audio
4. Program to hide the text file into the video

## **Task Remaining**

1. Program to encode the text file into the pdf files

2. A GUI interface for the program

## **Testing**

Using different test cases, test for each of the function developed until this period has been performed.

### **Testing table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Function | Test | Expected Result | Outcome | Evidence |
| 1. | Encryption | Generate the cipher text from the message file | A encrypted file containing the cipher text | Successful |  |
| 2. | Decryption | Decrypt the encrypted file | A text file with readable message | Successful |  |
| 3. | Encoding | Embed the message file in image | The message hidden in the image | Successful |  |
| 4. | Decoding | Decode carrier image to get hidden file | Recovered text file | Successful |  |
| 5. | Encoding | Embed the message file in audio file | The message file hidden in the audio | Successful |  |
| 6. | Decoding | Decode carrier audio to get hidden file | Recovered text file | Successful |  |
| 7 | Encoding | Encode the message file in video file | The message file hidden in the video | Successful |  |
| 8 | Decoding | Decode the carrier video file to get hidden message | Recovered | Successful |  |

### **Test evidence**

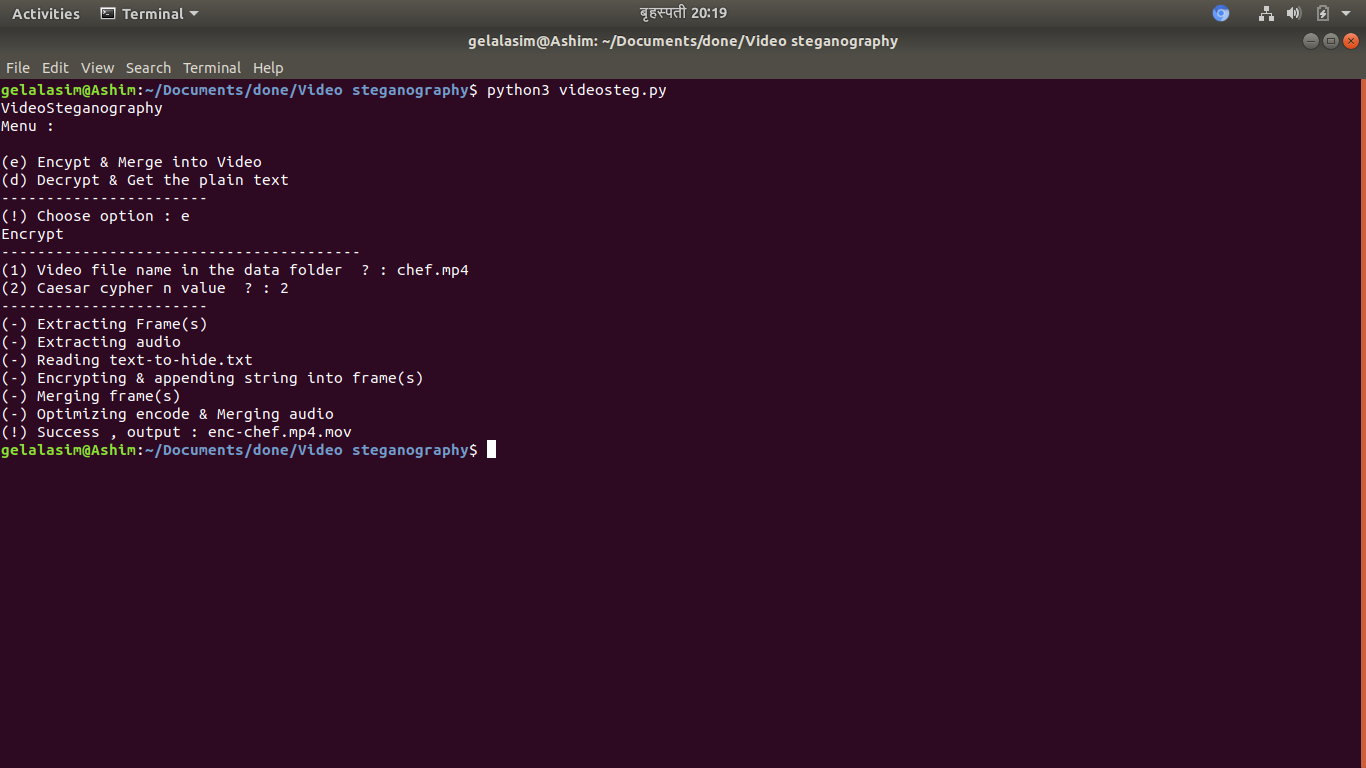
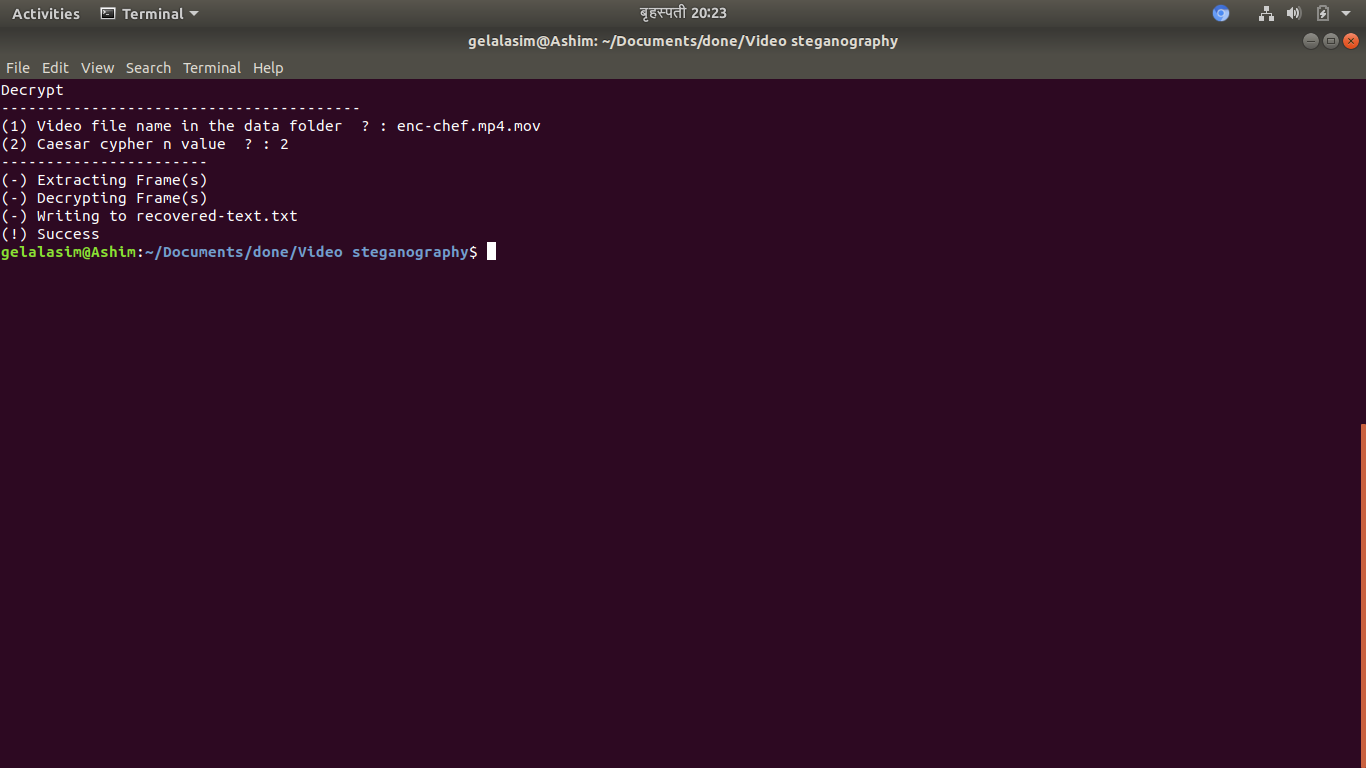
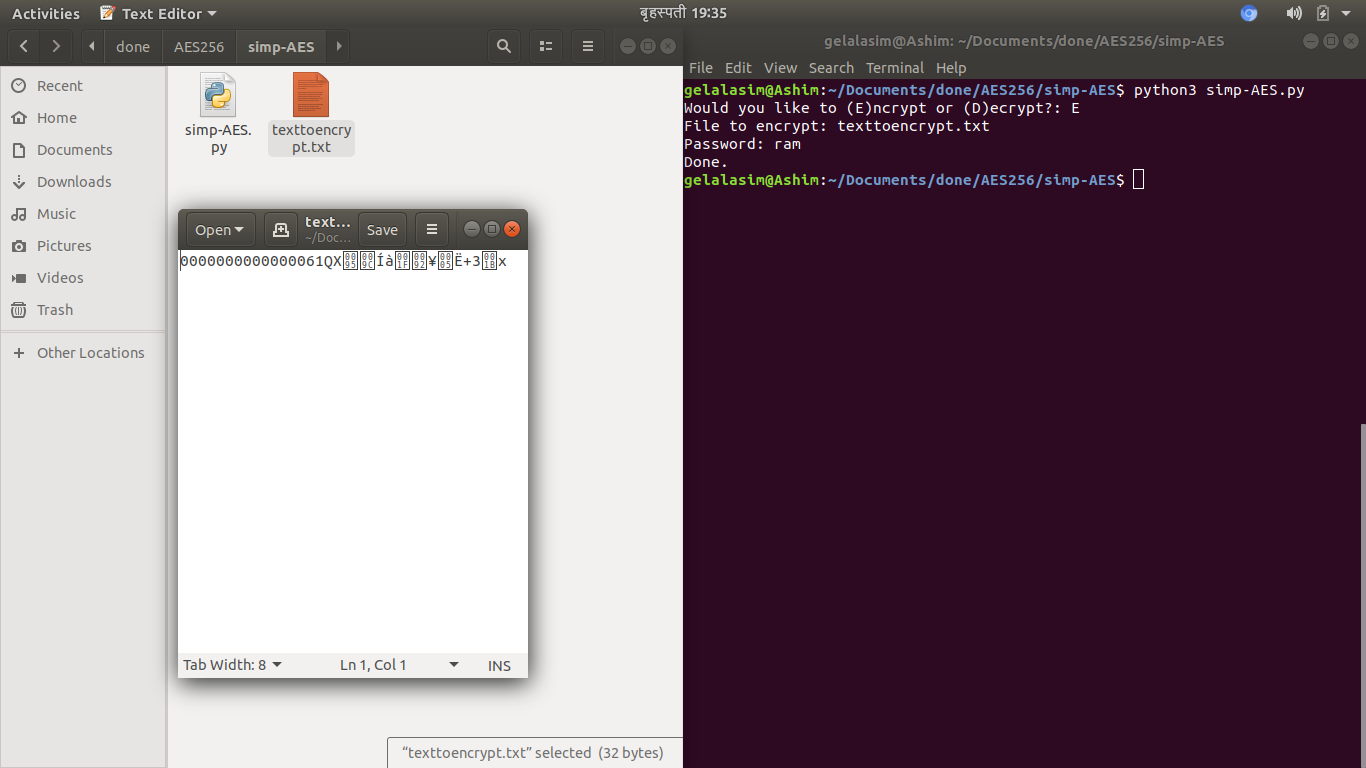
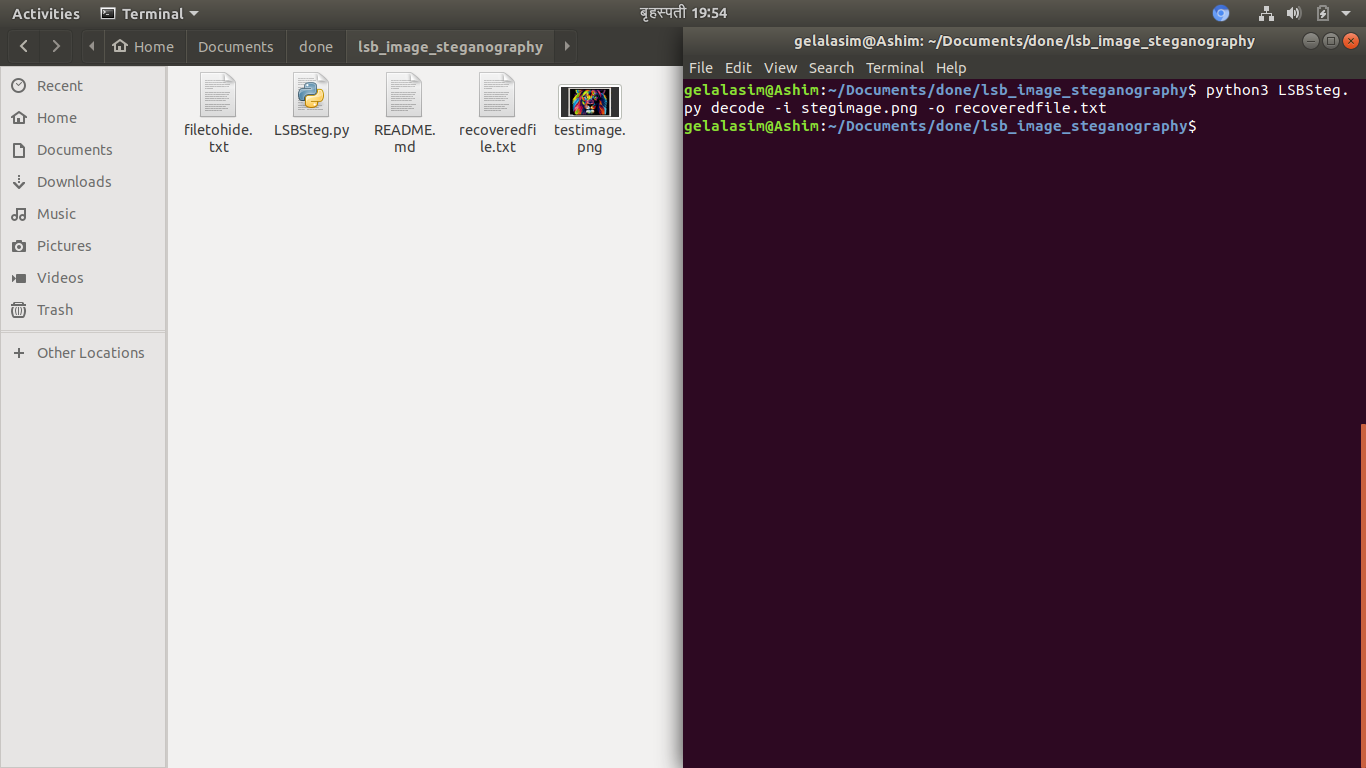
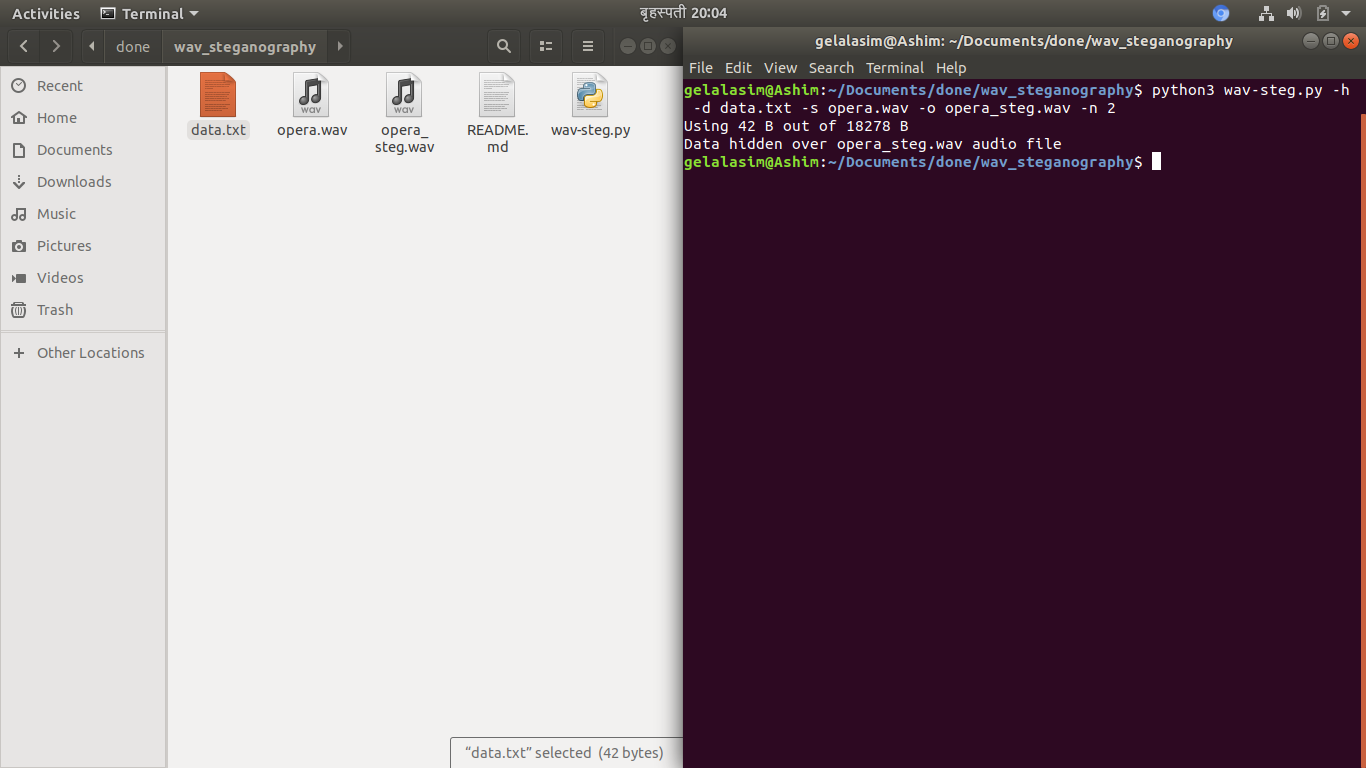


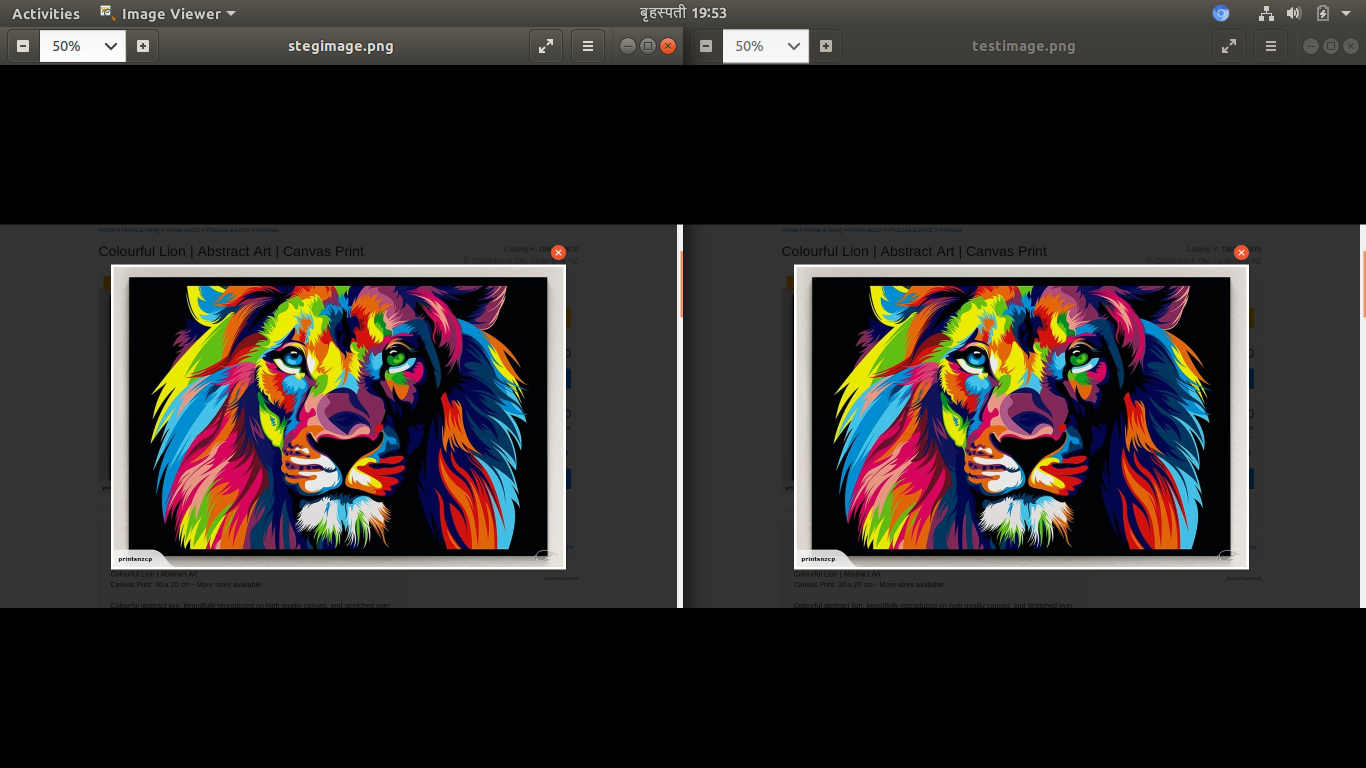
Figure 3.Sucessful video encoding

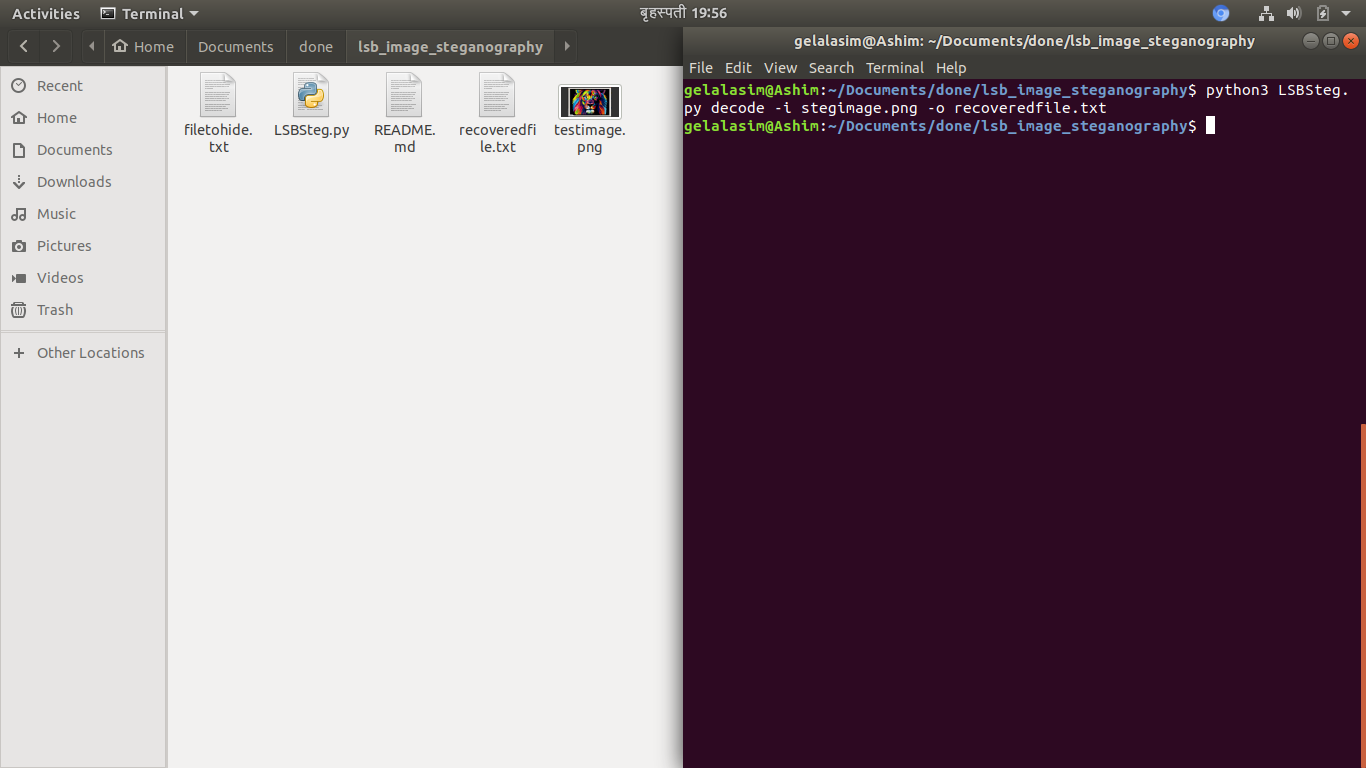


**Figure 4 Sucessful Video Decoding**









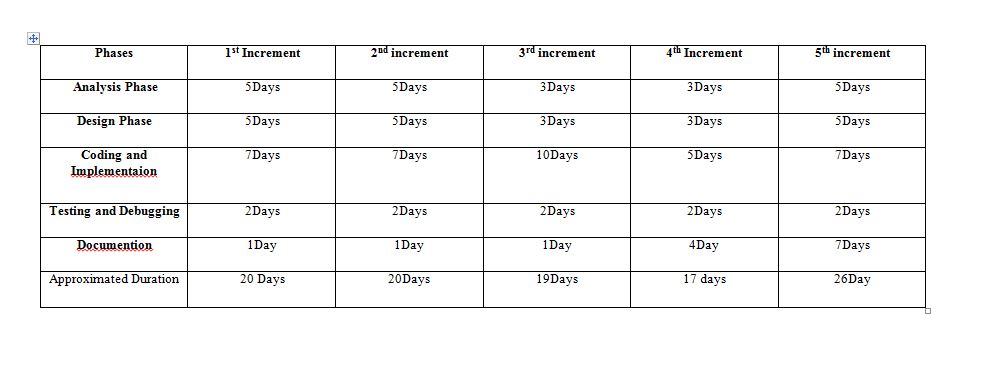
# DELIVERABLES

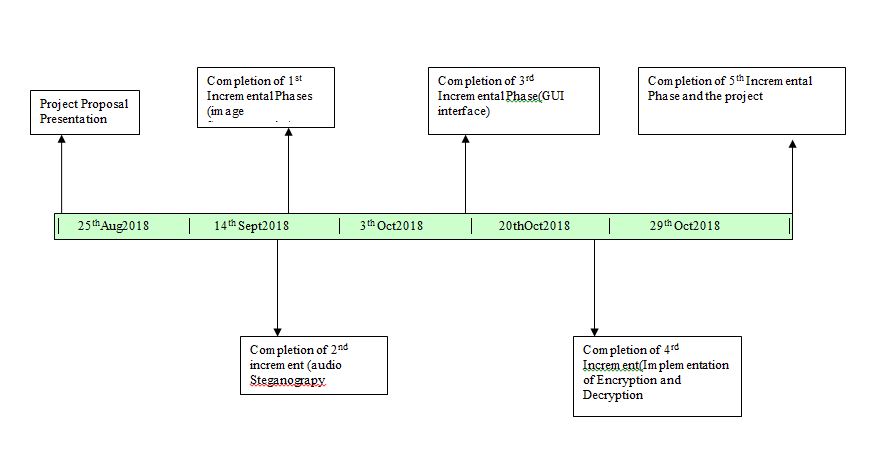
At the completion of this project, it will deliver the following:

1. A fully function desktop application program that can encrypt and decrypt the files, and also hide those encrypted file to a carrier media

2. A detailed project report document

# Task and Time Schedule





# BIBIOGRAPHY

1. Incremental Model, S. Pressman, Software Engineering Fundamentals, Seventh Edition.[offline]
2. Mike Driscoll: An Intro to encryption in Python3

[<https://dzone.com/articles/an-intro-to-encryption-in-python-3>[offline]

1. Image steganography [https://www.dreamincode.net/forums/topic/38678-image-steganography/[14 Aug,2018]
2. [LSB-DCT based Image steganography](https://stackoverflow.com/questions/35396977/lsb-dct-based-image-steganography)

[<https://stackoverflow.com/questions/35396977/lsb-dct-based-image> [steganography](https://stackoverflow.com/questions/35396977/lsb-dct-based-image-steganography)]

[14 Aug,2018]

1. Cryptography and Steganography with Python, [<https://opensourceforu.com/2010/05/cryptography-and-steganography-with-python/>][16Aug2018]
2. An Overview of Steganography James Madison University InfoSec Tech report Department of Computer Science

[<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.5129&rep=rep1&type=pdf>][20 Aug2018]

1. Best available Tools to hide data: [<https://www.geekdashboard.com/best-steganography-tools/#openstego>][21 Aug 2018]
2. Dr. Joyti Preet: Image steganography [<https://www.slideshare.net/hussainsavani/image-steganography>][22 Aug 2018]
3. Jayaram p,Rangaanatha H R, Anupama H S: INFORMATION HIDING USING AUDIO STEGANOGRAPHY–A SURVEY[]

[<http://aircconline.com/ijma/V3N3/3311ijma08.pdf>][22 Aug 2018]

1. Key Distribution for Symmetric Key Cryptography: A Reviewby Yashaswini J [<http://www.rroij.com/open-access/key-distribution-for-symmetric-keycryptography-a-review.php?aid=56128>][23 Aug 2018]
2. *AES-256 with random key generation instead of hash: [*[https://crypto.stackexchange.com/questions/50359/aes-256*-with-random-key-generation-instead-of-hash*](https://crypto.stackexchange.com/questions/50359/aes-256-with-random-key-generation-instead-of-hash)*][10sept 2018]*
3. *Python GUI – tkinter,[*[https://www.geeksforgeeks.org/python-gui-tkinter*/*](https://www.geeksforgeeks.org/python-gui-tkinter/)*]*