# SECURING DATA USING STEGANOGRAPHY AND CRYPTOGRAPHY

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

**Computer Science and Engineering**

By

Ayush Singh (171335)

Akanksha Choudhary (171351)

Under the supervision of

Dr. Vivek Sehgal

to

Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh**

# Candidate's Declaration

I hereby declare that the work presented in this report entitled **Akanksha Choudhary** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2021 to June 2021 under the supervision of **Dr. Vivek Sehgal** (Computer Science).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature) AChoudhary

Akanksha Choudhary, 171351

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature) Vivek Sehgal

Dr.Vivek Sehgal

Associate Professor

Computer Science

Dated: 18-05-2021

# Candidate's Declaration

I hereby declare that the work presented in this report entitled **Ayush Singh** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2021 to June 2021 under the supervision of **Dr. Vivek Sehgal** (Computer Science).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature) Ayush

Ayush Singh, 171335

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature) Vivek Sehgal

Dr.Vivek Sehgal

Associate Professor

Computer Science

Dated: 18-05-2021

# Acknowledgement

We would like to thank **Dr. Vivek Sehgal** for helping and guiding us throughout the semester in our project and supporting the concept.We have to put a lot of effort into this project, but this was not possible without the kind support and help of many people.So, we thank all of them sincerely. We are very grateful to our undertaking chief Dr. Vivek Sehgal for providing leadership and continuous supervision as well as the necessary information about our project and for supporting the completion of this undertaking. We would like to give special thanks and appreciation to our friends and colleagues who have given us time and attention.

Thankyou.

Date : 18-05-2021

Ayush Singh(171335)

Akanksha Choudhary(171351)

# Table Of Content

**Content** **Page No.**

# List of Abbreviations

| | | |
|---|---|---|
| **AES** | **-** | Advanced Encryption Standard |
| **BMP** | **-** | Bitmap |
| **CFB** | **-** | Cipher Feedback |
| **GIF** | **-** | Graphic Interchange format |
| **JPEG** | **-** | Joint photographic experts group |
| **LSB** | **-** | Least Significant Bit |

# List of Figures

# List of Tables

# Abstract

One of the foremost important concern with the web users who are transmitting highly secure information over internet is secure data transfer, with the spread of computerized information round the globe through the internet, the safety of the data has raised a worry to the final population.

So, to beat this issue we includes the steganographic techniques. Using steganographic senders can hide their secure data within the kind of image, audio, and video files.

The receiver needs all the shares to recover the pictures with hidden message then the message to be decoded from the image.

The main objective behind this project is to transmit a message on a channel where another quite information is already being transmitted.The project is worried with the applying development to secure your data by converting your data in UTF-8 char codes

then embedding it in a picture so data is shielded from the intruders. The system makes the information double secure by embedding it in a picture and usage of RGB and UTF char codes. In this project user will choose a picture and convert the secret data into cipher text by using cryptography techniques and then embed the data into that image and at the receiver side the cipher text is decrypted from the image and by using decryption algorithms convert that cipher text into the original message.

# Chapter 01 : INTRODUCTION

## 1.1 Introduction

- While transfering a file from one point to a different through internet we’d like file secure concepts.

- This project helps us to send a file from one place to a different in a very secured manner.

- The software we employed in our project are:-
  - For backend-

    Jupyter Notebook

- Programming languages we used:-
  - Python

- By the algorithms which is predefined by the user. It is one in numerous information concealing techniques that transmit message over channel wherever another normal data is already being transmitted.

- Steganography means hiding the messages in such an easy way where nobody aside the user and receiver, will check the existence.

### 1.1.1 Encryption and Decryption

Encryption is said to be a process which transforms the initial information into an unrecognizable form.

At the encryption side we'll first take plain text and by using the encryption algorithm it will convert it into the cipher text.

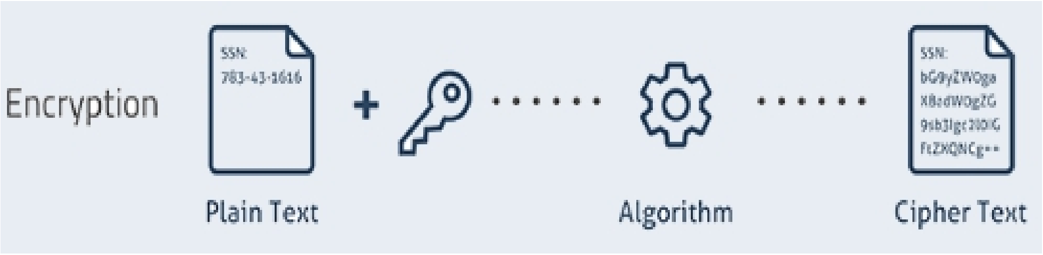


Fig.1.1 Encryption Process

Decryption said to be a process of converting encoded/encrypted data in an exceedingly form that's readable and understood by somebody's or a computer.

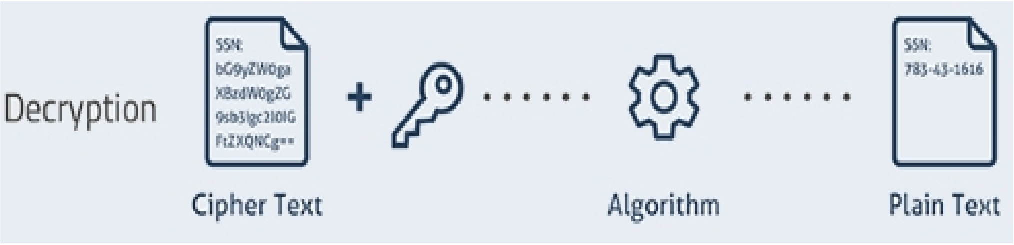


Fig.1.2 Decryption Process

### 1.1.2 Steganography and Cryptography

Cryptography may be a process which secures data using some algos in order so only that people for which the knowledge was meant will know it and process it.

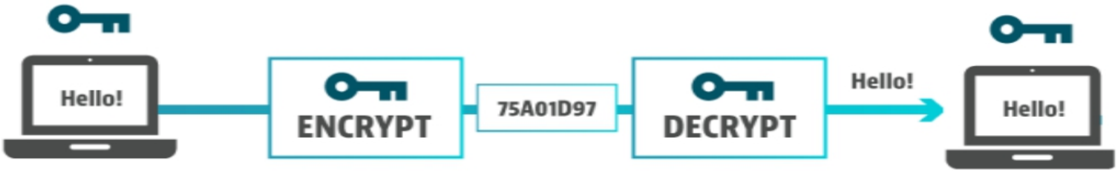


Fig.1.3 Cryptography

Steganography is one of many information hiding method, which aims to transmit the message over to the user where different kinds of data is send.

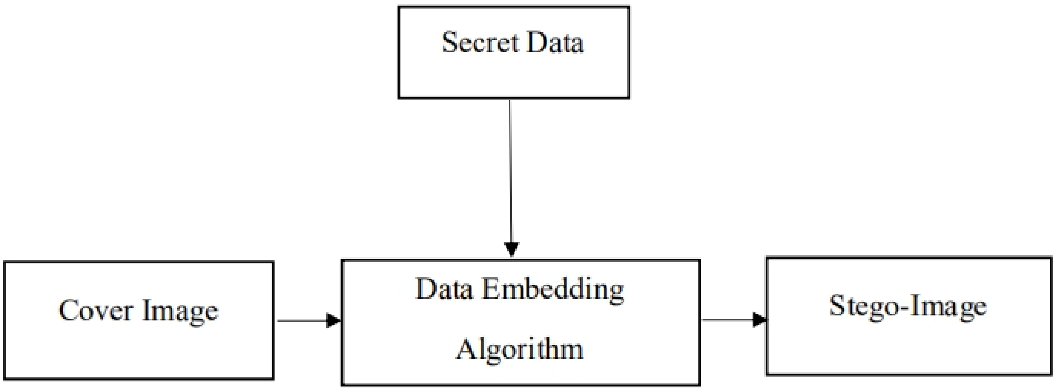


Fig.1.4 Steganography

The above figure shows that if a message is inserted in a picture through an algo, with the help of some other data or key. The result is stegnographic picture which send to other side where it’s analysed by the decryption algo.

Stego-image :- Stegnographic image is that which is the output of the process of embedding. Stego images have the messages hidden in either the values of pixels or in some optimally selected coefficients.

### 1.1.3 Some Modern Cryptography algorithms

- **Block Ciphers :**

  Block cipher consists for both algorithm that is encryption and decryption:

  1. First we give a key (K) to the sender and other text, and their product consists of cipher text block which is denoted by C.
  2. D is the decryption algo which is opposite of earlier algo in which cipher text is converted to original.

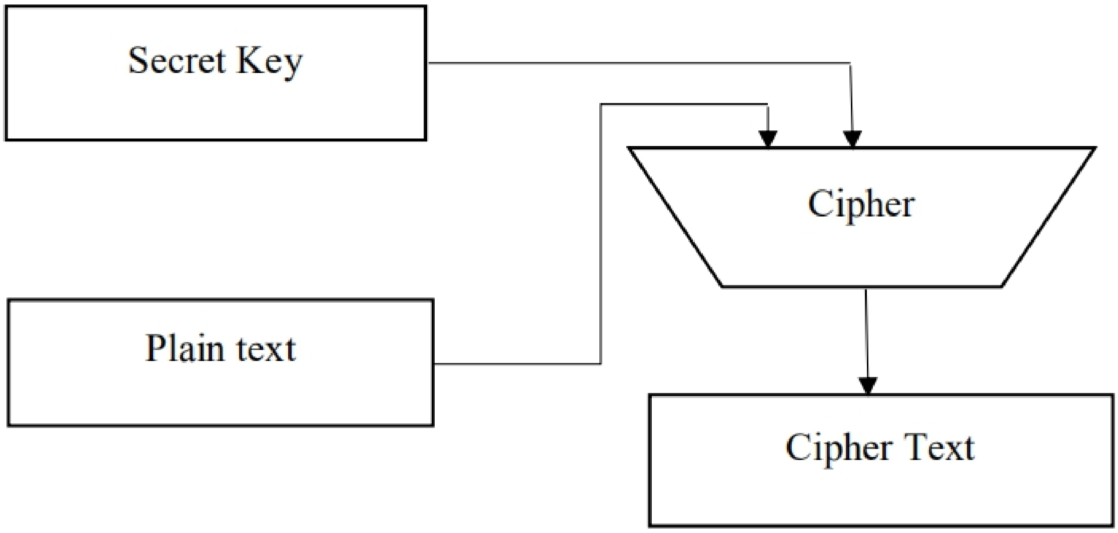


Fig.1.5 Block cipher diagram

- **Stream Ciphers :**

  It will treat pseudo random bits given from key, and therefore the plain text is converted by perform some operation with both the plain text and also the pseudo random bits. This type of cypher were sometimes ignored within the past.

  Below figure shows the asyn- and sync types.

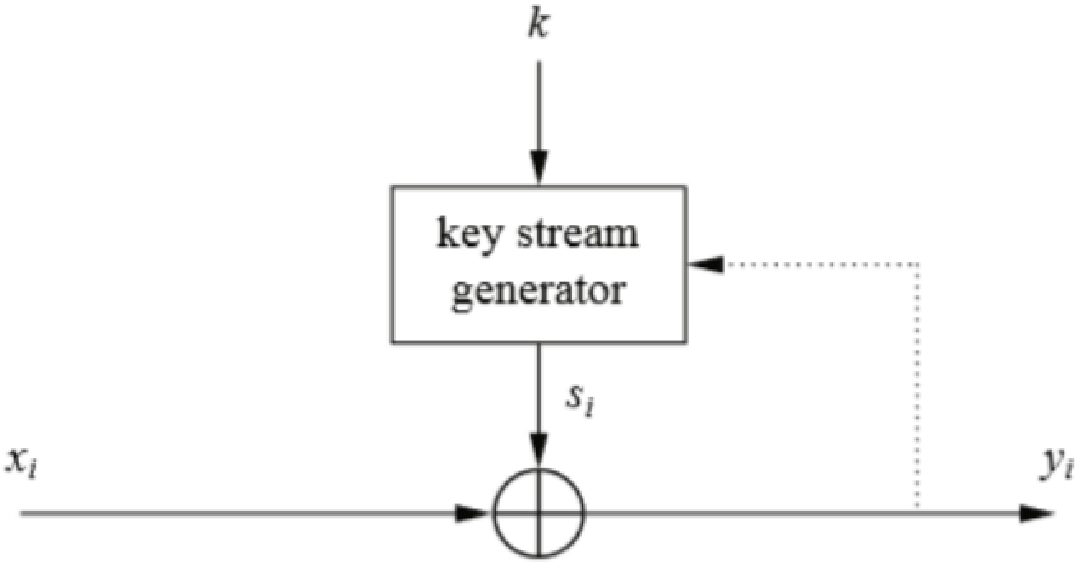


Fig.1.6 Asynchronous and synchronous types of stream ciphers

- **Hash Functions :**

  Earlier their is a function,which worked by maping by taking random input to a stucked output in an exceedingly process which is called compression, This process is not identical because the compression done in zip.

i. **Pre image collision resistance**: This type of property will generate when the input find a way to hash to give an output.

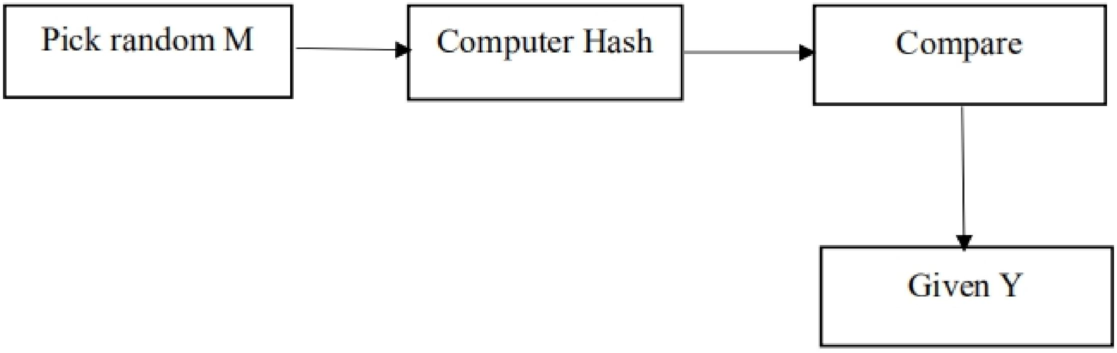


Fig.1.7 Preimage collision resistance

ii. **Second pre image collision resistance:** This is another way which is shown in below picture.

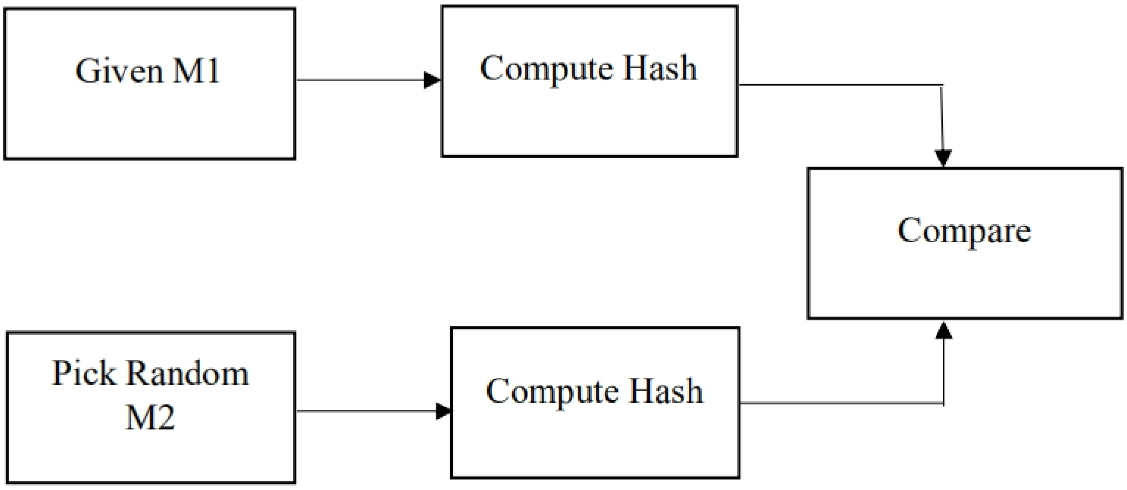


Fig.1.8 Second Preimage collision resistance

- **Digital Signature :**

Unlike cryptography, digital signatures didn’t exist before the invetion of computers. The requirement arose for digital sgnatures to be discussed, especlly within the business environments where multiple parties happen and every must attempt to keeping their declarations and/or proposals. The subject of unforgotable signatures was first discussed centuries ago, except those were handwritten signatures. The though behind digital signatures was first

A. **Digital Signature Requirements :**

The connection b/w which created the link between sign and encryption came into present with the “digitalization” era in which we were currently living and aalso become the witness of that.

B. **Digital Signature Principles :**

The below picture will the show the process of this method:-

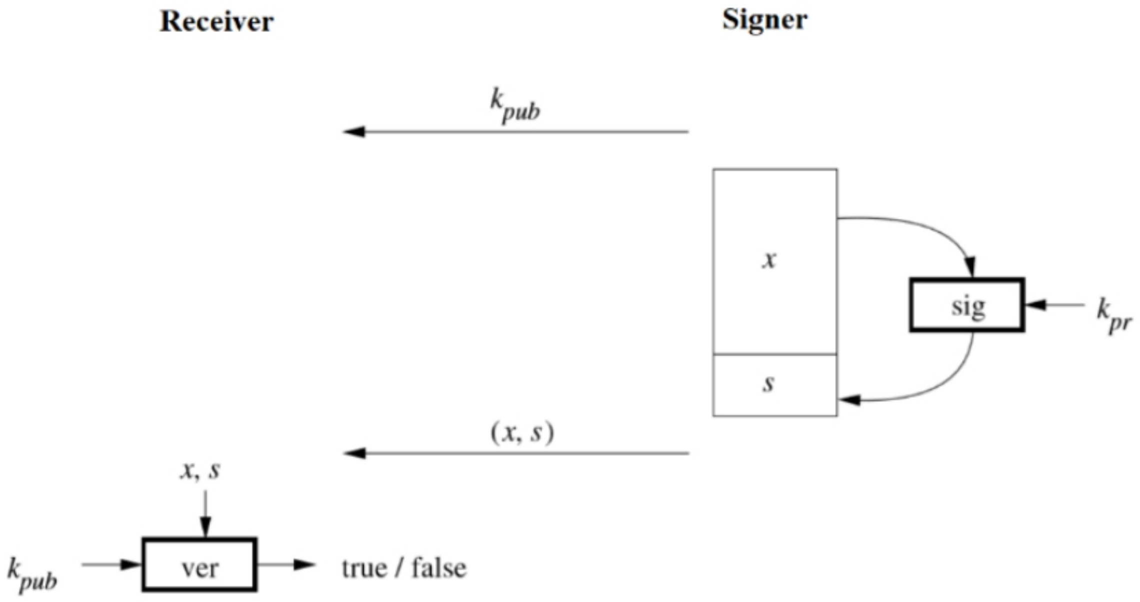


Fig.1.9 Digital Signature Principal (signing and verifying)

The function will require an external key so as order to link the signature to the sender who signed it, and also it become output of the verification function would be either "true" or "false". The output would be true in an exceedingly case during which the msg was signed through the private key thats linked with the opppositr key, the public verification key. Otherwise the output of the verification perform would be false.

### 1.1.4 Some Steganographic definitions :-

Below mentioned are some defintions which include :

**Cover medium :** his could be the another way to hide the knowledge.

**Embedded message :** It refers to a message which is secrectively enter inside a picture.

**Stego-medium :** It's often be often the last word piece of data that the can be seen.

### 1.1.5 Types of Steganography :-

- **Secret key Steganography :**

  Secret key steganography is outlined as a steganographic system that needs the exchange of a secret key (stego-key) before communication only the persons who know this secrete key can reverse the method using stego-key and skim the message. The benefit to Secret Key Steganography is whether or not it's intercepted, only parties who know the key can extract the key message.

- **Pure Steganography :**

  Pure steganography is defined as a steganographic system where there's no must exchange any password that's stego-key so as for the receiver to read the message. This method of Steganography is that the least secure means by which to speak secretly because the sender and receiver will bank solely upon the presumption that no alternative parties are responsive to this secret message.

### 1.1.6  Image Steganography :-

Because the name suggests it refers to the strategy of concealing information among all information. Picture chosen for this purpose is termed **cover-image** and therefore thepicture generated after this process is termed the **stego-image**.

Digital pictures those who seem on your laptop are jerky into pixels little dots with a selected color that along structure the image you ll be ready to see. For pictures steganographers code the message into the picture element lsb. This suggests that to the human eye the colour of the element diagram-matical by code to the pc doesn't change. The hidden msg was taken from the image provided something that know a) that there's a msg within the picture b) that you just use the identical steganographic program for decoding because the one want to hide the message.

Within this reason a text ciphertext alternative pictures or something which can be entered in a veryn exceedingly are going to be hidden in a picture. This process has come back quite so much in recent years with the event of quick graphical things and steganographic code is currently without delay on the market over Infobahn for everyday users.

### Concealment in digital images :

Their are many processes by using which the data can be inserted in a picture, but we were using the below approach.

Common approaches include:

- **Least significant bit (LSB) insertion :**

  The right corner important bit insertion methodology is maybe the foremost renowned image steganography technique. It’s a typical straightforward approach to insert a data. Unfortunately, it’s extremely oberserved by attacks, such as image manipulation a straight forward conversion from a GIF format another format like jpeg.

  In below we’ll show how first 3 pixels will be 3 twenty four bit words below:

  (00100111 11101001 11001000)

  (00100111 11001000 11101001)

  (11001000 00100111 11101001)

  Letter A is having binary conversion is 10000011 which will bw ranging from highest left but:

  (0010011111101000 11001000)

  (00100110 11001000 11101000)

  (11001000 0010011111101001)

- **Masking and Filtering :**

  To create a watermked picture, we add the lumnance of the masked area by fifteen percent. If we were to vary the luminance by a smaller percentage, the mask would be undetected by the human eye. Now we will use the watermarked image to cover plain text or encoded information.

  Masking is more robust than LSB insertion with relevancy compression, cropping and some image processing. Masking techniques embed information in additional significant areas so the hidden message is more integral to the duvet image than simply hiding it within the "noise" level.

- **Algorithms and Transformations :**

  To using redudant picture , you'll want to trade off the size of msg opposite to the robustess. An out sized message may be embedded just an occasion because it might occupy a far greater portion of the image area.

  Apart from this another methods will perform some operations and scatter the hidden data throughut a picture. Scattering the msg makes it appear more like noise.

  Scattring and encrypt will support to help against hidden msg to extract it from the picture but not against message destruction through image processing. A scattered message within he image's LSBs continues to be as prone to destruction from lossy compression and image processing, as may be a clear text message inserted within the LSBs.

### 1.1.7 Implementation of Steganography

Below picture will show that we will first a data then by using encrypting algo we'll convert that data into cypher text which will be used to insert in a picture which will be result in output image which is send over a channel to another side which is a receiver which extract that cypher text from picture and by detecting which algo will be used to convert data into cypher it will convert that data into original msg.

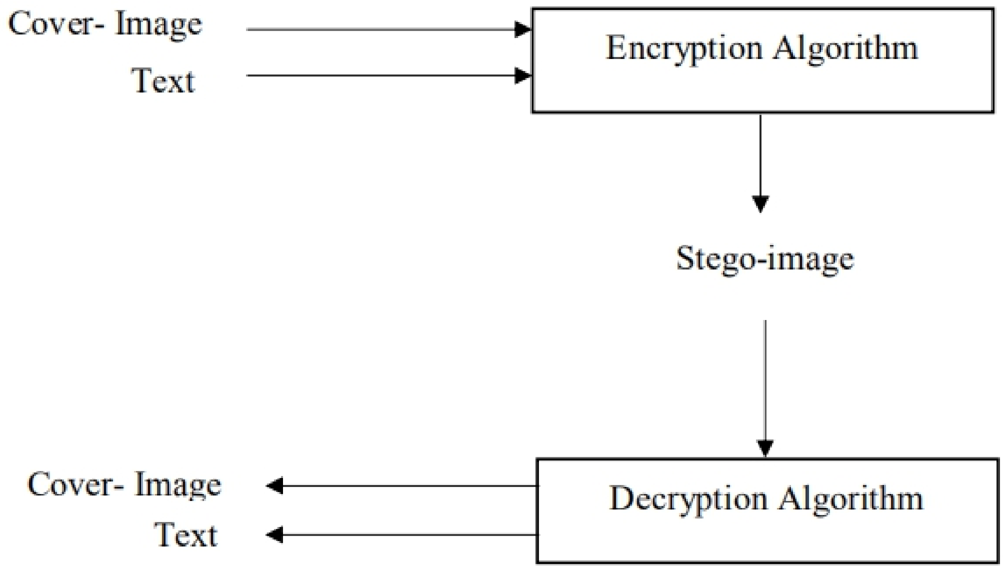


Fig.1.10 Process of Image Steganography

## 1.2 Problem Statement

Computer becomes more complicated and sophisticated, security becomes a major factor. Many securing data methods require several other methods. Secure transmissions are put in site to stop attacks like good spoofing and general data loss. Hence, in order to produce an improved mechanism, we are using RGB and UTF char codes during this report we will be using Steganography technique that’s to embed our data into a picture in order that no outsider is attentive to the presence of knowledge.

Steganography has improved greatly in recent years, as many digital methods now allow hiding of data inside other information in new ways, & it could be importat in several conditions.

Steganography may be utilized in an outsized value of information formats in the present generation. Some of the data formats are .doc, .gif, .jpeg, .bmp, .wav, .txt and .mp3. Primarily due to their demand over the web and this method will use easy techniques. These formats are popular due to the relative ease by which redundant or noisy data are often far from them and replaced with a hidden message. Steganographic technologies are a awfully important a part of the longer term of Internet security and privacy on open systems like the Internet.

So, as to produce a steganography platform. We will convert our original data into cipher text and embed that cipher text into the picture by using LSB technique and at the receiver side we will decrypt the data from the picture and using the cryptography decryption algorithms convert that cipher text into the original message.

## 1.3 Objectives

The project has the subsequent objectives:

- To form an application which will be want to hide information.
- Authorization is needed to use the applying.
- The applying should be easy.
- The applying ought to encipher and decode the info.
- The applying provides enough security for your confidential information.

## 1.4 Methodology

Two pictures, key-image and encrypted-text-image, that are slightly different from each other, can be wont to calculate variations between their pixels which may be regenerate into utf char code which implies text.

### Data Hiding

Data concealment actual helpful information is scattered throughout the image and caps are full of random trash. Rgb that consists of three separate numbers are often accustomed hide knowledge well from any algorithmic program that searches for similarities between a pair of or additional footage that were geerated by a similar key the char code will be randomly distributed between the red, green and blue which will be added or subtracted from the original rgb. Therefore, if the same file is used to encode same data, still it will look different from previously made pictures and actual data pixels are undifferentiable from "trash" data, which also will change in a random way every time.

## 1.5 Organization

- In Chapter 1, Introduction and basic idea used in designing the project is mentioned. Objectives of the project, methodology exists in this chapter.

- In Chapter 2, Literature review is given, includes various research papers on Cryptography and Steganography, used to compare our results with the existing ones.

- In Chapter 3, System Development is discussed which includes software configuration and hardware configuration, front end and back end applications and their features.

- In Chapter 4, Performance Analysis is shown with the helps of screenshots of the application developed to ensure data security.

- In Chapter 5, Conclusion and Future scope of the project is mentioned.

# Chapter 02 : LITERATURE SURVEY

Encryption and secret writing a sort of hiding methods refers to the method of stealing data so the detector cannot sight the data.

In arithmetic applied science associated connected an formula is a good technique for finding a drag consits of several instructions. Algorithms are used widely for processing information and in plenty of different fields.Each algorithmic rule may be a list of well outlined directions for finishing a task.Starting from Associate in Nursing initial state the directions describe a computation that payoff through a well outlined series of ordered states eventually terminating in a very final ending state.The transition from one state to succeeding isn't essentially deterministic.Some algorithms called randomized algorithms incorporate randomness.

## 2.1 Cryptography

In this project, we will take some of the cryptographic techniques. First of all in our project their is a option given to the user which is chosen a algo to encrypt the data, so the user will encrypt the data according to them. We were using two methods i.e. Caesar cypher and AES encryption algorithm.

The first one will encrypt the data by shifting the alphabets where in case of second one it will having a advantage to choose a key. We will send the same key to other side so that if sender will choose AES method to encrypt the data then the other side will also choose the same to detect the original data.

## 2.2 Steganography

After converting our data to cypher text we use LSB technique to put our cypher text inside the picture. Basically we will take 500 x 500 x 3 size of our image and in this we will first convert our data into binary bits then each pixel is having rgb values we will take LSB bit of each pixek and replace it with our data bi.

Also we will add five hashes at the end so that when we send our data to receiver then its algo will retrieve that data from picture which also includes five hashes then algo will get to know that whenever five hashes will come it remove that hashes and remaining part will be our data.

Also at the end of the data their is a number written which shows that which algo would you to use to encrpyt the data then receiver's algo would get to konw and use the same algo and detect the original data.

**In this project we use following steganography :**

- Video Steganography
- Image Steganography
- Document Steganography
- Audio Steganography

Their are several types but we were using Image Steganoraphy for our project.

| METHODS | WEEK POINTS |
|---|---|
| Least Significant Bit | 1- Simple to extract<br>2- Compression can damage the data |
| Parity writing | Straightforward to extract and destroy |
| Echo concealing | Low inserting capability and security |
| Phase coding | Low capacity |
| Spread spectrum | 1-Requires more space.<br>2- Unprotected to time scale modification |
| Wavelet Domain | Retrive data might have lossy |

Table.2.1 Existing systems and their weak points

This paper [12] found out an image stego method that enhances the existing LSB substitution techniques techniques improve the protection level of hidden data and increase embedding capability of hidden data. Lossless compression technique are going to be utilised to compress secret data and hash operate is employed to hash the hidden information. Hash operate are going to be dead within the stego image and its values will be hold on in the hostimage for more checking throughout extraction process. The planned technique demonstrates considerably improvement in terms of knowledge security embedding capability and quality.

# Chapter 03 : SYSTEM DEVELOPMENT

## 3.1 Technology Used

### 3.1.1 Hardware Configuration

- Processor: 2.3 GHz Intel® Core™ or equivalent
- Memory: 2 GB

### 3.1.2 Software Configuration

- Operating System: Windows XP or higher
- Language: Python
- Framework: Jupyter Notebook

## 3.2 Jupyter Notebook

Jupyter Notebook and Jupyter Lab is an open source web based Integrated development environment for scientific computing and data science, but it can be used to quickly prototype almost any python based software.

## 3.3 Python

Python is one of the most popular programming language, which is high-level object oriented, which has dynamic semantics, and is interpreted. It also has some high-level built in data structures. It has dynamic binding and also typing. It is an extremely readable and intuitive language. With the huge number of useful libraries freely available for python, it is very fast to use and to prototype applications. Programmers also prefer python due to the increase productivity that comes by using it, because it is an interpreted language, which means that it does not have an extra step of compiling. Moreover, testing and debugging is very easy.

**Few benefits of python are:**

- It has vast community
- It has huge number of free libraries
- It is open source
- It is very secure
- It can work on any platform
- It is very readable and simple to understand
- It is simple to write
- Codes are generally concise
- Quick prototyping, due to interpreter based
- Python can work in procedural execution, oop as well as with functions.
- It can handle big data
- It can perform complex maths.

There are numerous uses of python, and programmers find newer ways to use python all the time. Few of the common uses of python are in Web development, Software dev, Data Science, Maths, Machine Learning, Artificial Intelligence, System Scripting.

## 3.4 Use Case Diagram

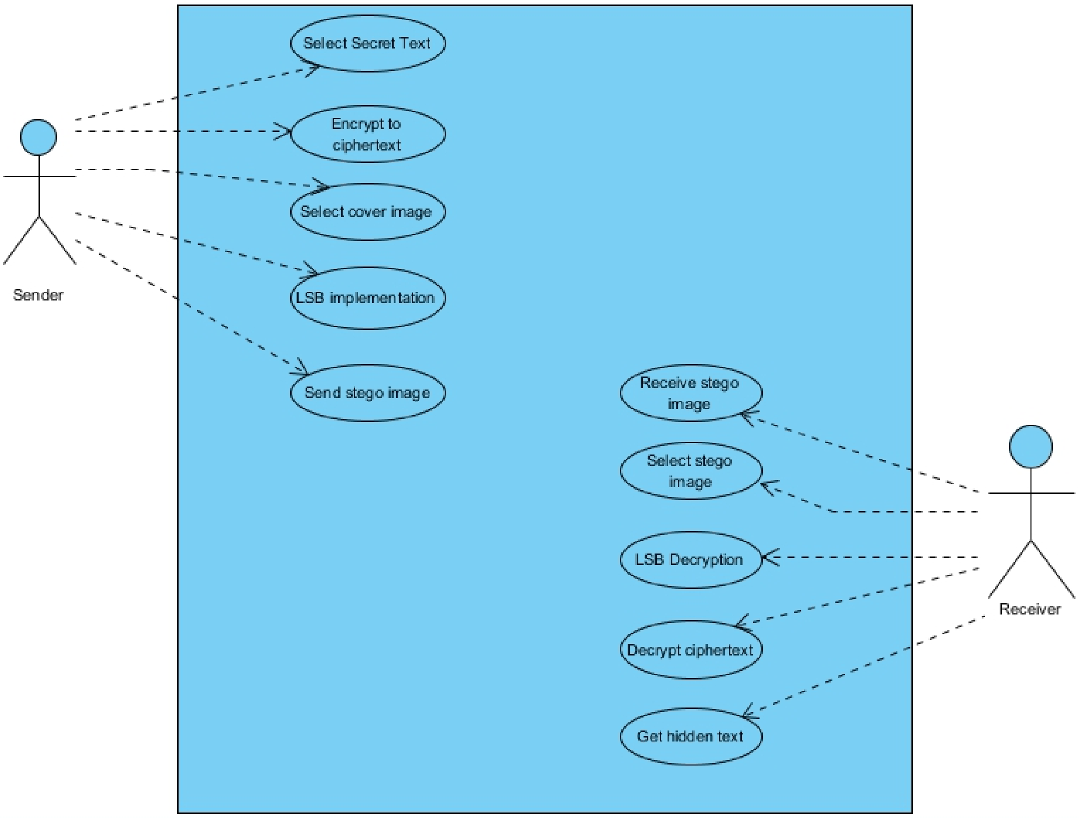


Fig.3.1 Use Case Diagram

### 3.5 Data Flow Diagram

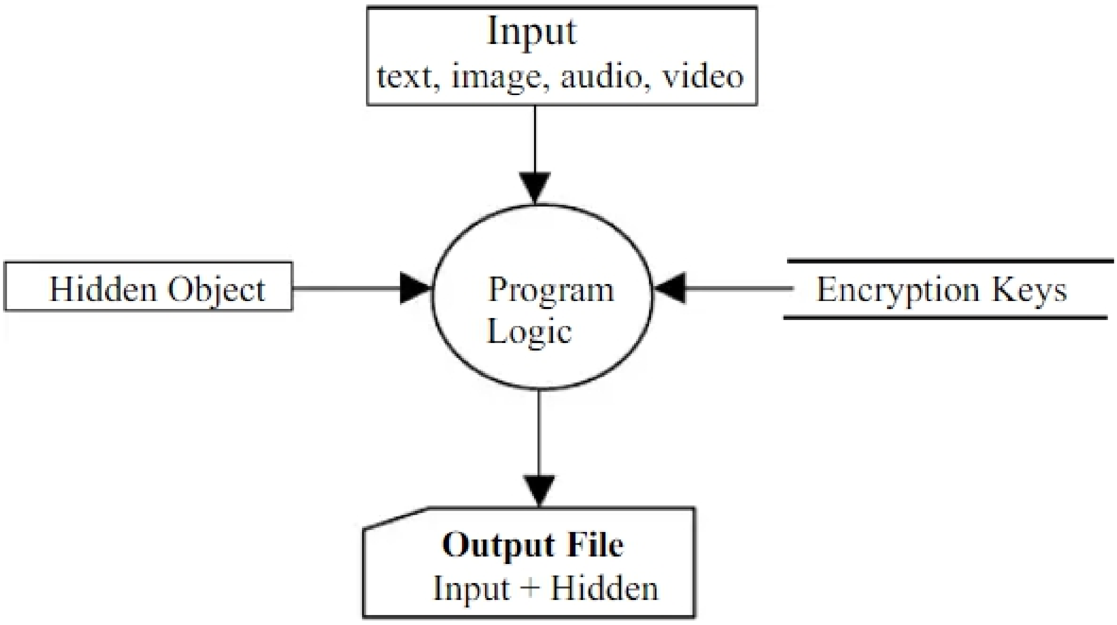


Fig.3.2 Data Flow Diagram

### 3.6 Comparison of Proposed and existing methods

Some Existing methods are not secure enough so here are some points in Table 3.3 which compares our proposed method with the existing ones.

| **Steganography** | **Existing Algorithms** |
|---|---|
| It provides double security to your data | Single level security is provided |
| Probability of detecting the hidden data is less | As, the data is shuffled on the basis of applied algorithm it can be cracked easily |
| RGB algorithm changes only least bit of the pixel which do not create any visible change in the image | Algorithm is applied to whole data so if encrypted data is visible one can easily find way to decrypt it |

Table.3.1 Proposed and Existing methods

## Chapter 04 : PERFORMANCE ANALYSIS

- **Home Page**

When we run the project, first window appears as shown in (Fig 4.1). In this window, we have two buttons:

**LOG-IN :** User can direct login into application.

**SIGN-UP :** New user can make their own account and can use the application.

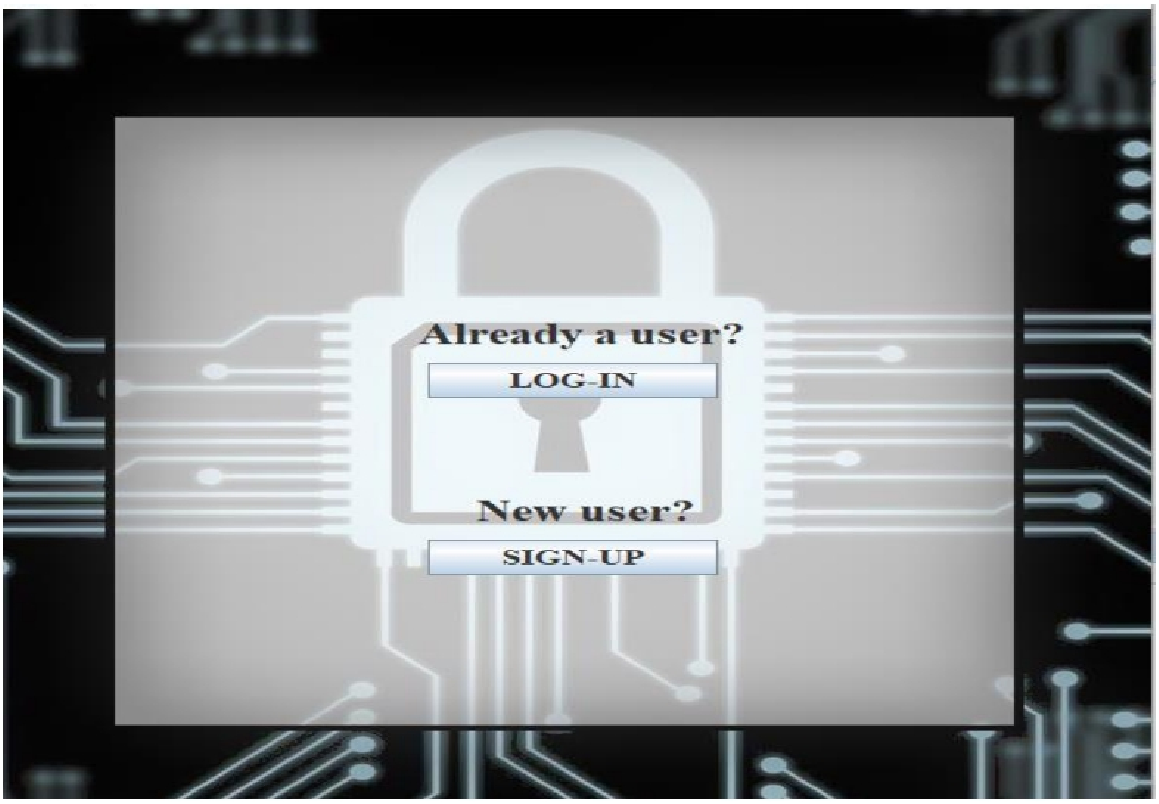


Fig.4.1 Application Home Page

- **Login Page**

When you click on the LOG-IN Button appears on the Window, it has 3 buttons:

**LOG-IN :** Opens the LOGIN page again.

**LOGIN :** User will go to the next Steg page.

**Back :** User will go the Home Page.

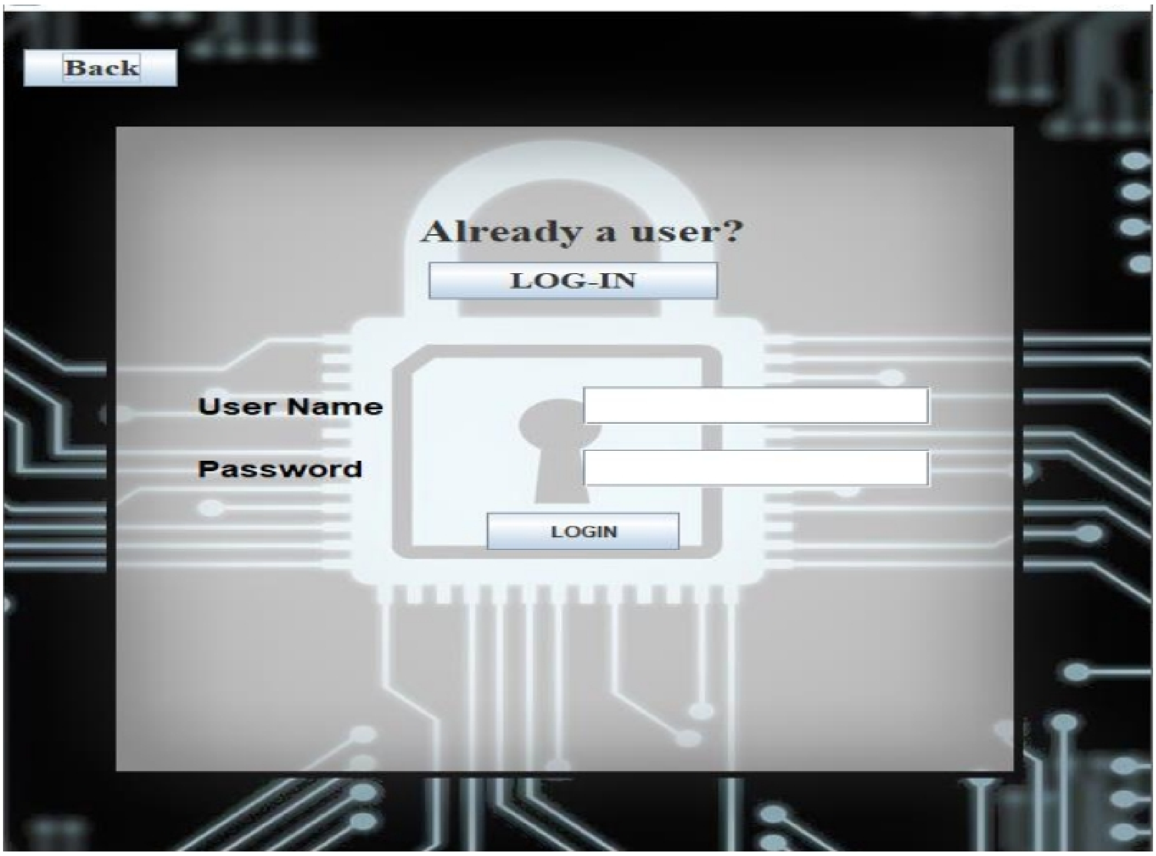


Fig.4.2 Login Page

- **Sign-Up Page**

When you click on the SIGN-UP Button appears on the Window, it has 3 buttons :

**SIGN-UP :** Opens the SIGN-UP page again.

**Back:** User will go the Home Page.

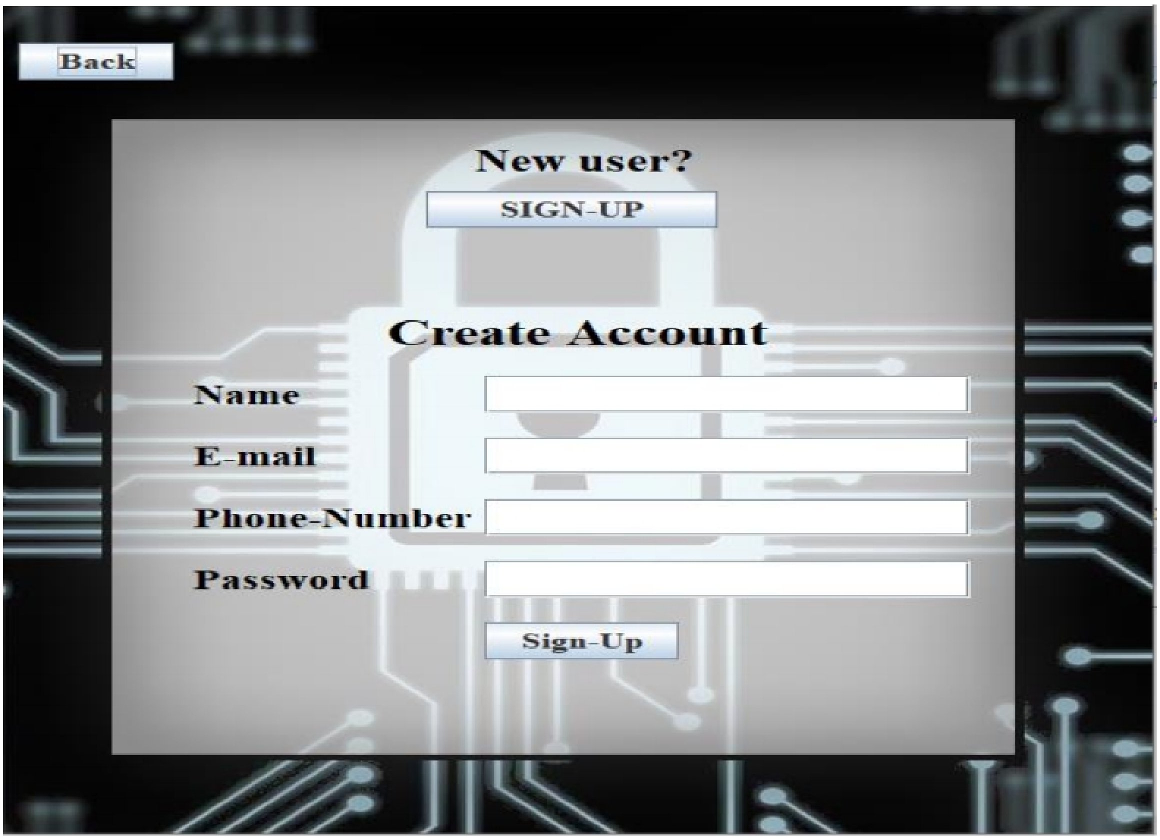


Fig.4.3 Sign-Up Page

- **Encryption & Decryption Choice**

Once user has Logged-In into the Application (Fig. 4.4) appears in the Window, it has two buttons, and a label with your User name:

**Encrypt:** User can encrypt his data.

**Decrypt:** User can decrypt his data.

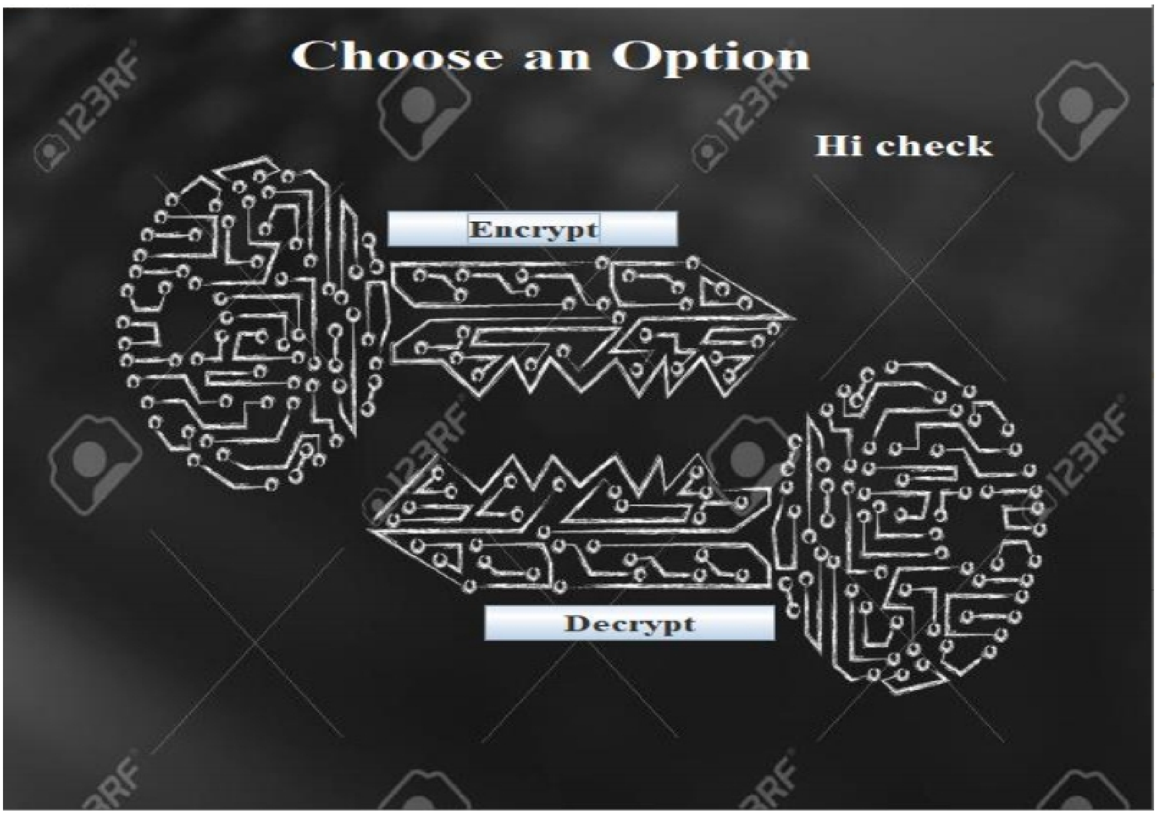
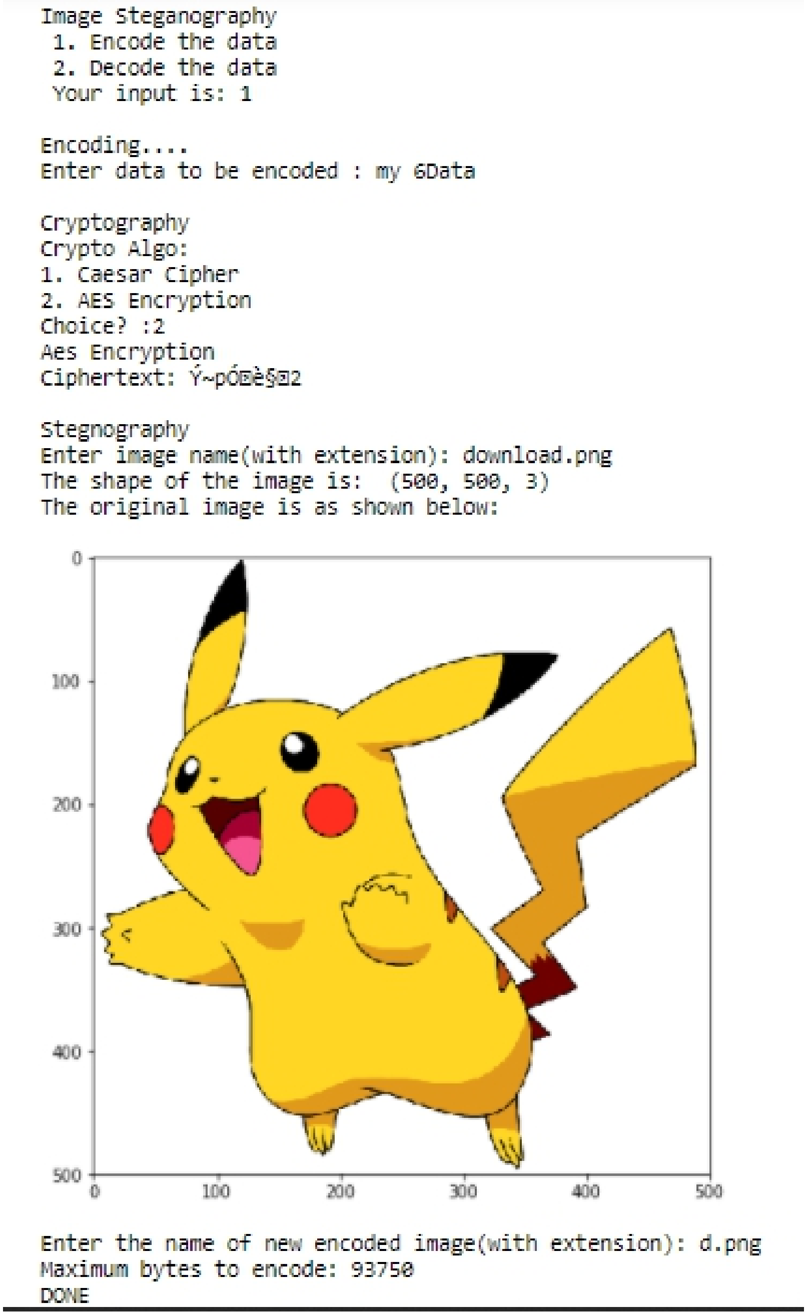


Fig.4.4 Encryption & Decryption Choice

- **Encryption**



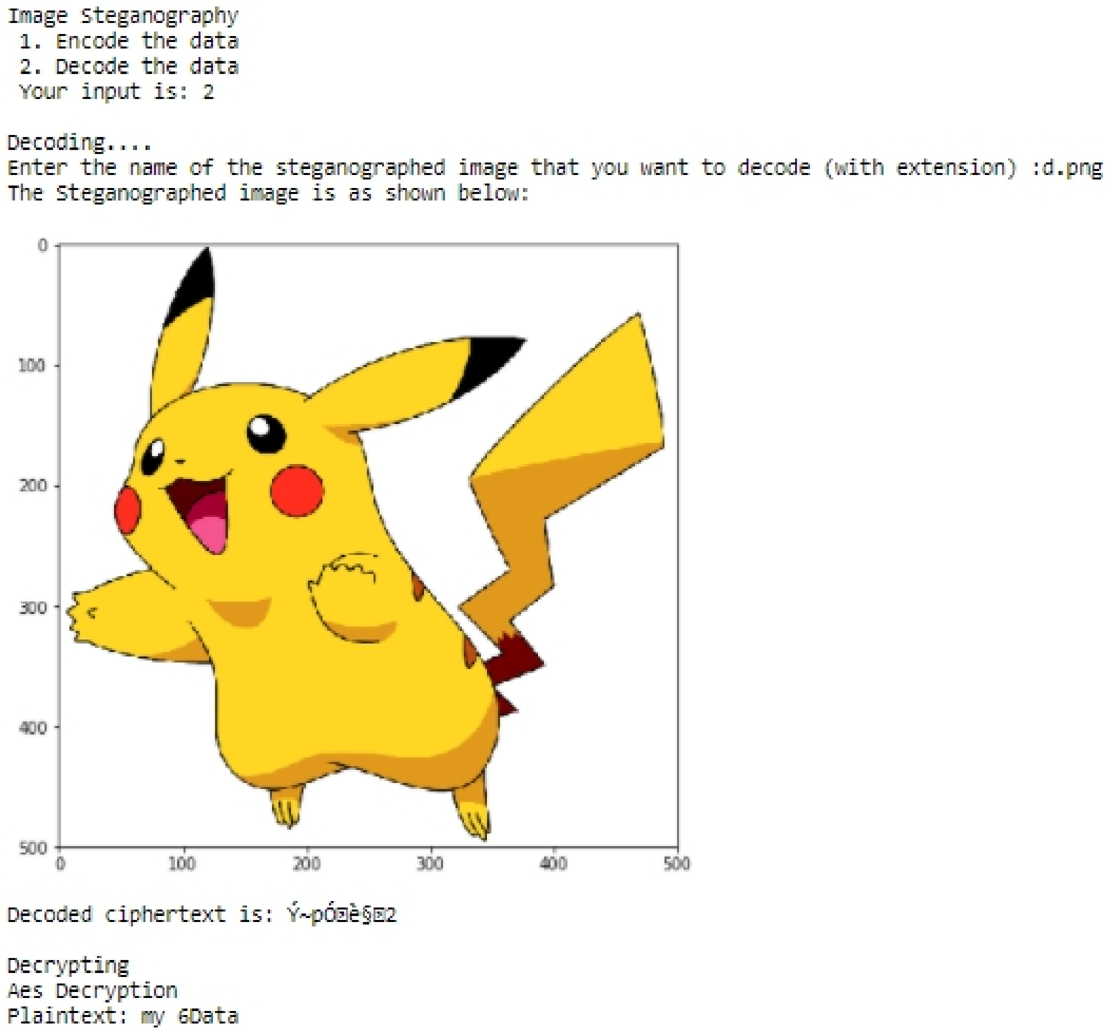
```
Image Steganography
 1. Encode the data
 2. Decode the data
 Your input is: 1

Encoding....
Enter data to be encoded : my 6Data

Cryptography
Crypto Algo:
1. Caesar Cipher
2. AES Encryption
Choice? :2
Aes Encryption
Ciphertext: Ý~pÓ⊡è§⊡2

Stegnography
Enter image name(with extension): download.png
The shape of the image is:  (500, 500, 3)
The original image is as shown below:
```

```
Enter the name of new encoded image(with extension): d.png
Maximum bytes to encode: 93750
DONE
```

Fig.4.5 Encrypted Image

Secret data is converted into the cipher text, then cipher text and image is input by the user and that text is encoded into the image and stego image is the output.

## ➢ Decryption



```
Image Steganography
 1. Encode the data
 2. Decode the data
 Your input is: 2

Decoding....
Enter the name of the steganographed image that you want to decode (with extension) :d.png
The Steganographed image is as shown below:
```

```
Decoded ciphertext is: Ý~pÓ⊠è§⊠2

Decrypting
Aes Decryption
Plaintext: my 6Data
```

Fig.4.6 Decryption

Stegano graphed image name is entered by the user and it will give the decrypted cipher text and by using decryption algorithm convert that cipher text into decrypted message.

# Chapter 05 : CONCLUSIONS

## 5.1 Conclusion

Recent advances in the field of Data Security have opened the way for the practical implementation of various types of Encryption and Steganography methods. After going through several paper, it was found that for implementing Data security there are various encryption algorithms which are not providing satisfactory security. So, this project is designed for implementing the steganography technique on a very basic RGB algorithm which was found to be more efficient than other basic algorithms.

## 5.2 Future Scope

**This project can be further upgraded in following areas:**

- To use more than one encryption algorithm and let give users option for selecting any one encryption algorithm for encrypting the file.
- To make application, use a public and private key to hide and extract the data.
- To make application which support video, audio embedding.
- To make application which support all formats for embedding.
- To make this application as a stand-alone application.

## 5.3 Applications

- Hiding a message with steganography strategies reduces the prospect of a message being detected.
- The benifit steganography has over cryptography techniques, by itself is that messages do not attract any suspicion to themselves.
- Difficult to detect, only if receiver has prior knowledge of presence of secret message.
- Can be used on any digital image or audio or video file with sufficient size.

## References

- Johnson, N. and Jajodia, S., “Exploring Steganography: Seeing the Unseen”,IEEE, Volume: 31 , Feb. 1998, pp.26-34.
- C. Kurak, J. McHugh, "A Cautionary Note On Image Downgrading", *Proc. IEEE Eighth Ann. Computer Security Applications Conf.*, pp. 153-159, 1992.
- TanmyBhaowmik, PramathaNathBasu, “On Embedding of text in Audio – A case of Steganography”, International Conference on Recent Trends in Information, Telecommunication and computing, IEEE 2010.
- W. D. A. M. E. HELLMAN, "New directions in cryptography," IEEE Transactions on Information Theory, Vols. IT-22, no. 6, pp. 644-654, 1976.
- S. J. Lincke and A. Hollan, "Network Security: Focus on Security, Skills, and Stability," in 37th ASEE/IEEE Frontiers in Education Conference, Milwaukee, 2007.
- H. Orman, "Recent Parables in Cryptography," IEEE Internet Computing, vol. 18, no. 1, pp. 82-86, 2014.
- R. GENNARO, "IEEE Security & Privacy," IEEE Security & Privacy, vol. 4, no. 2, p. 64 - 67, 2006.
- B. Preneel, "Cryptography and Information Security in the Post-Snowden Era," in IEEE/ACM 1st International Workshop on TEchnical and LEgal aspects of data privacy and SEcurity, Florence, 2015.
- Johnson, N. and Jajodia, S., “Exploring Steganography: Seeing the Unseen”,IEEE, Volume: 31 , Feb. 1998, pp.26-34.
- C. Kurak, J. McHugh, "A Cautionary Note On Image Downgrading", Proc. IEEE Eighth Ann. Computer Security Applications Conf., pp. 153-159, 1992.
- B. Preneel, "Cryptography and Information Security in the Post-Snowden Era," in IEEE/ACM 1st International Workshop on Technical and LEgal aspects of data privacy and security, Florence, 2015.

# Appendices

- **Algorithm**

**Algorithm for embedding data inside image :**

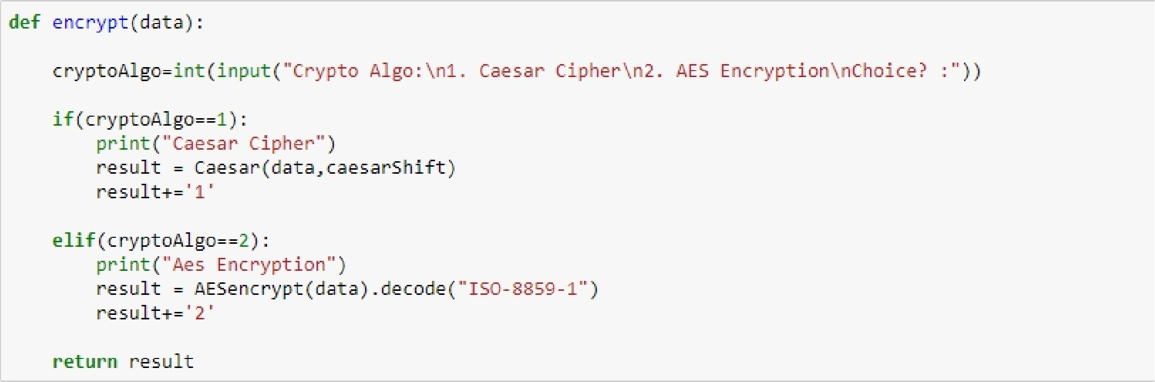
```
Convert Secret_Message to Cipher_text;
Input:    Cover_Image, Cipher_text;
      Convert Secret_Message to bits;
      Imbed Bit_Secret_Message to Cover_Image pixels;
Output: Stego_image;
```

**Algorithm for extracting data from stego image :**

```
Input : Stego_image;
            Decode all Bit_Secret_Message form Stego_image by each pixel;
            Convert Bit_Secret_Message to Cipher_text_Message;
            Convert cipher_text_Message to Secret_Message;
Output : Secret_Message;
```

## ➢ Code

**Convert data into cipher text :**

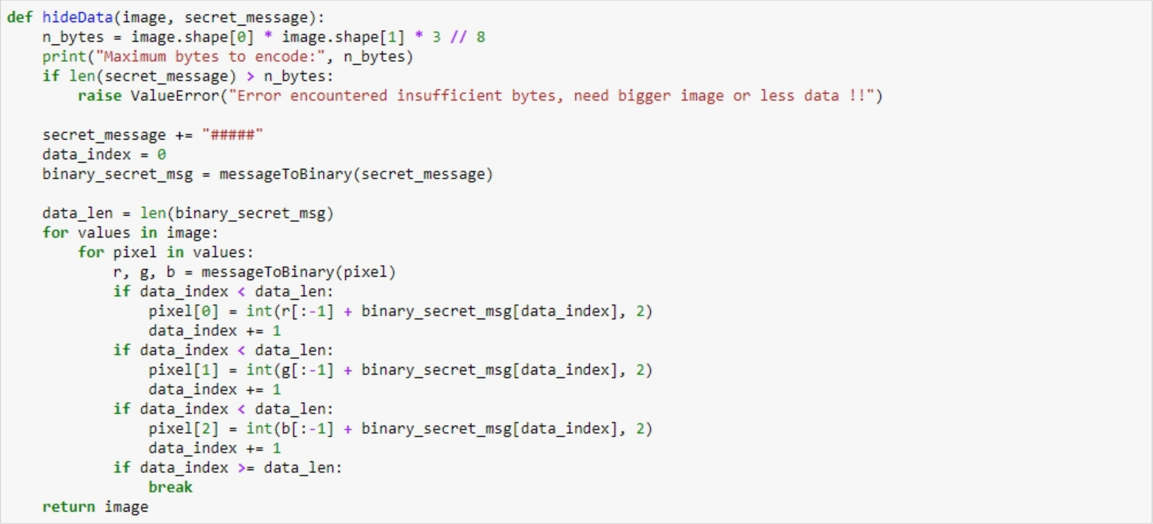


```
def encrypt(data):

    cryptoAlgo=int(input("Crypto Algo:\n1. Caesar Cipher\n2. AES Encryption\nChoice? :"))

    if(cryptoAlgo==1):
        print("Caesar Cipher")
        result = Caesar(data,caesarShift)
        result+='1'

    elif(cryptoAlgo==2):
        print("Aes Encryption")
        result = AESencrypt(data).decode("ISO-8859-1")
        result+='2'

    return result
```

In this piece of code we'll do cryptography. Basically, by asking the user that which algo they want to use to convert the original text into the cypher text.

Also by doing we assign that part to the result variable and at the end we will add some number which will help the receiver program to find out which algo was used to make cypher, so that they can recover the original msg sent by the sender.

### Hide cipher text into image :

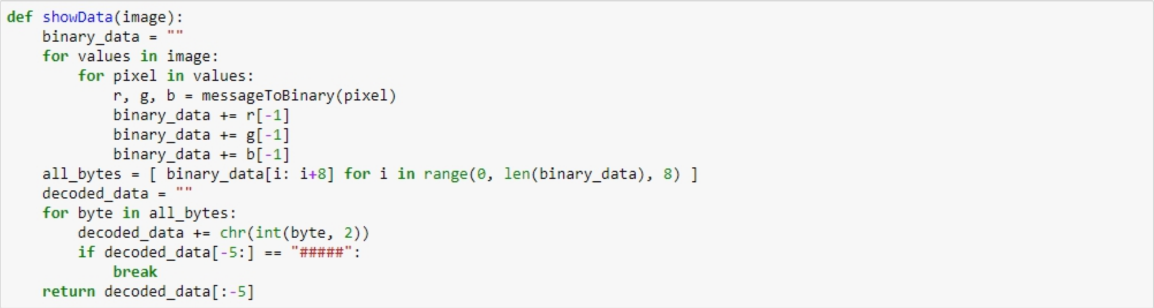


```
def hideData(image, secret_message):
    n_bytes = image.shape[0] * image.shape[1] * 3 // 8
    print("Maximum bytes to encode:", n_bytes)
    if len(secret_message) > n_bytes:
        raise ValueError("Error encountered insufficient bytes, need bigger image or less data !!")

    secret_message += "#####"
    data_index = 0
    binary_secret_msg = messageToBinary(secret_message)

    data_len = len(binary_secret_msg)
    for values in image:
        for pixel in values:
            r, g, b = messageToBinary(pixel)
            if data_index < data_len:
                pixel[0] = int(r[:-1] + binary_secret_msg[data_index], 2)
                data_index += 1
            if data_index < data_len:
                pixel[1] = int(g[:-1] + binary_secret_msg[data_index], 2)
                data_index += 1
            if data_index < data_len:
                pixel[2] = int(b[:-1] + binary_secret_msg[data_index], 2)
                data_index += 1
            if data_index >= data_len:
                break
    return image
```

This function reads this picture, prints its shape, converts the msg into binary and then adds “#####” at the end of msg, so that the receiver can figure out when the msg is terminated.

This function then converts the data into the picture red, green and blue pixel values.

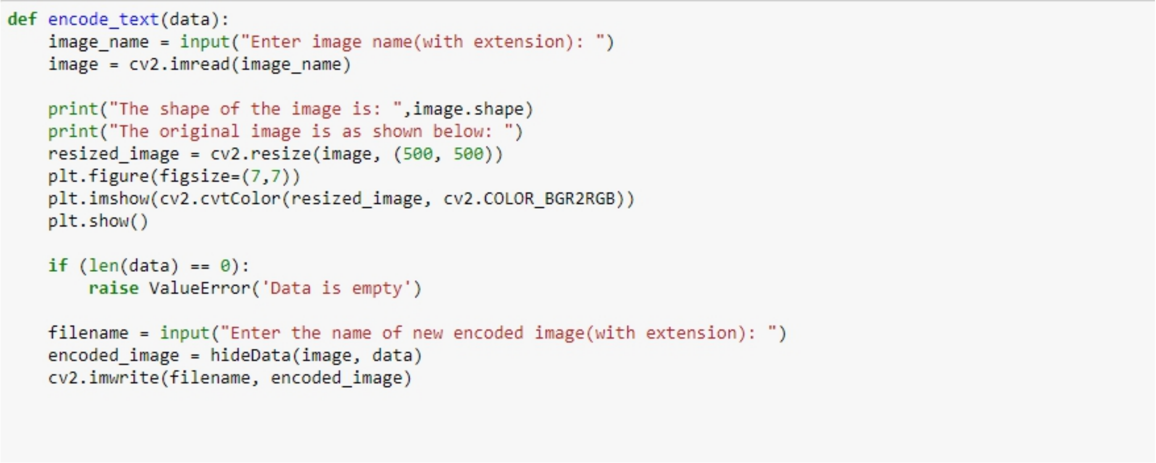
### Extract hidden cipher text from image :



```
def showData(image):
    binary_data = ""
    for values in image:
        for pixel in values:
            r, g, b = messageToBinary(pixel)
            binary_data += r[-1]
            binary_data += g[-1]
            binary_data += b[-1]
    all_bytes = [ binary_data[i: i+8] for i in range(0, len(binary_data), 8) ]
    decoded_data = ""
    for byte in all_bytes:
        decoded_data += chr(int(byte, 2))
        if decoded_data[-5:] == "#####":
            break
    return decoded_data[:-5]
```

The above code extracts the data from picture red, green and blue pixel values and stops reading the data when “#####” are encountered and then removes the hashes and returns the resulting data.

**Input and pre-process data before encoding :**

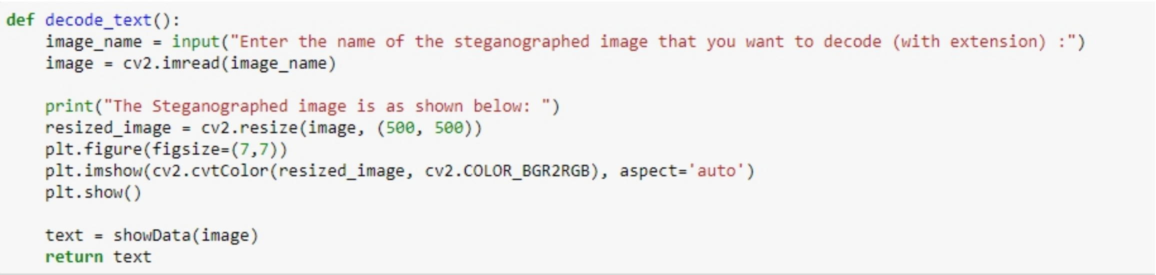


```
def encode_text(data):
    image_name = input("Enter image name(with extension): ")
    image = cv2.imread(image_name)

    print("The shape of the image is: ",image.shape)
    print("The original image is as shown below: ")
    resized_image = cv2.resize(image, (500, 500))
    plt.figure(figsize=(7,7))
    plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
    plt.show()

    if (len(data) == 0):
        raise ValueError('Data is empty')

    filename = input("Enter the name of new encoded image(with extension): ")
    encoded_image = hideData(image, data)
    cv2.imwrite(filename, encoded_image)
```

This function reads the picture from intrenal storage in same folder and then processes it and sends it to ‘hide data function’.

This function then saves this resulting picture from RAM back to intrenal storage in same folder as a new file.

### Input image to decode, and display hidden message :

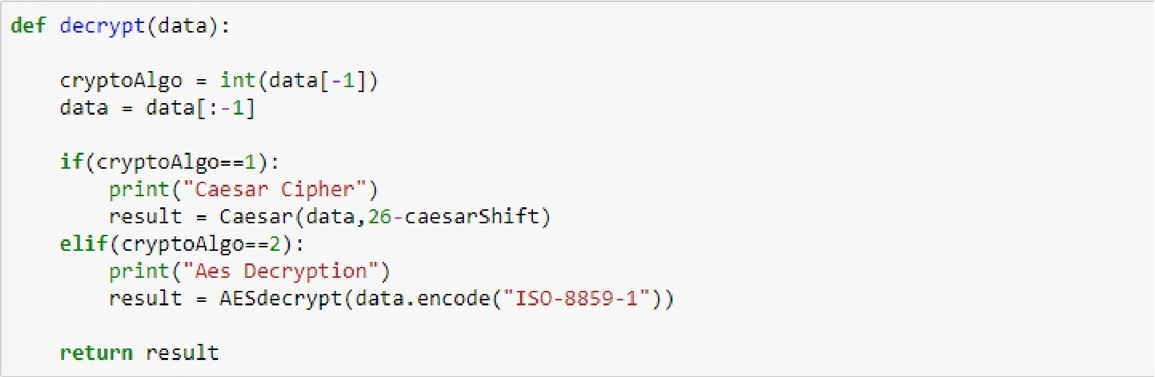


```
def decode_text():
    image_name = input("Enter the name of the steganographed image that you want to decode (with extension) :")
    image = cv2.imread(image_name)

    print("The Steganographed image is as shown below: ")
    resized_image = cv2.resize(image, (500, 500))
    plt.figure(figsize=(7,7))
    plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB), aspect='auto')
    plt.show()

    text = showData(image)
    return text
```

This function reads the resulting picture from intrenal storage in same folder and then processes it and sends it to ‘show data function’.

This function then returns the resulting data to the other side and discards the picture in RAM.

### Convert hidden message(cipher text) into original data :

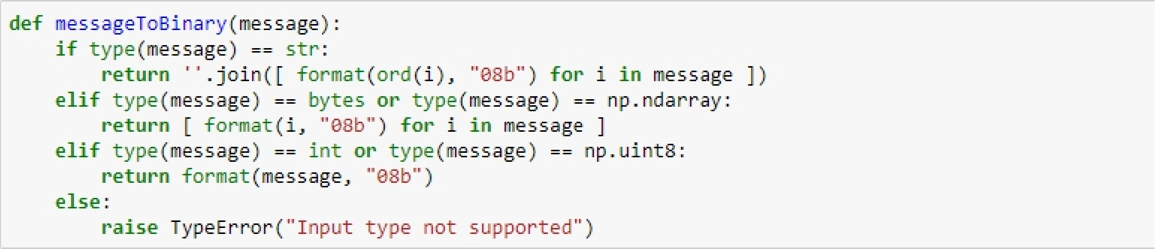


```
def decrypt(data):

    cryptoAlgo = int(data[-1])
    data = data[:-1]

    if(cryptoAlgo==1):
        print("Caesar Cipher")
        result = Caesar(data,26-caesarShift)
    elif(cryptoAlgo==2):
        print("Aes Decryption")
        result = AESdecrypt(data.encode("ISO-8859-1"))

    return result
```

This function receives cypher text along with a last bit which shows that what crypto algo was used.

This function then uses that algorithm to convert the cypher into original value and then it returns to the other side.

### Converting input to binary:

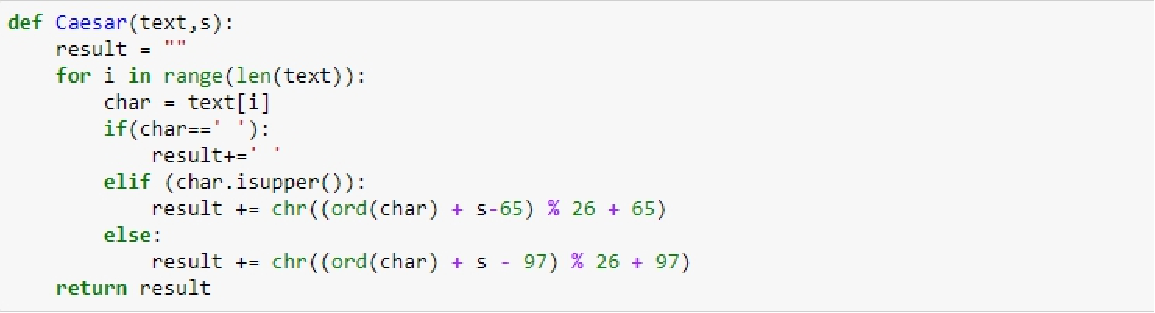


```
def messageToBinary(message):
    if type(message) == str:
        return ''.join([ format(ord(i), "08b") for i in message ])
    elif type(message) == bytes or type(message) == np.ndarray:
        return [ format(i, "08b") for i in message ]
    elif type(message) == int or type(message) == np.uint8:
        return format(message, "08b")
    else:
        raise TypeError("Input type not supported")
```

This function receives an input from multiple finctions and detects the type of input data and then converts it into binary data and returns back this binary data to the calling function.

This function also supports string type data, byte typr data, array type data and integer typr data.

### Casear Cipher:

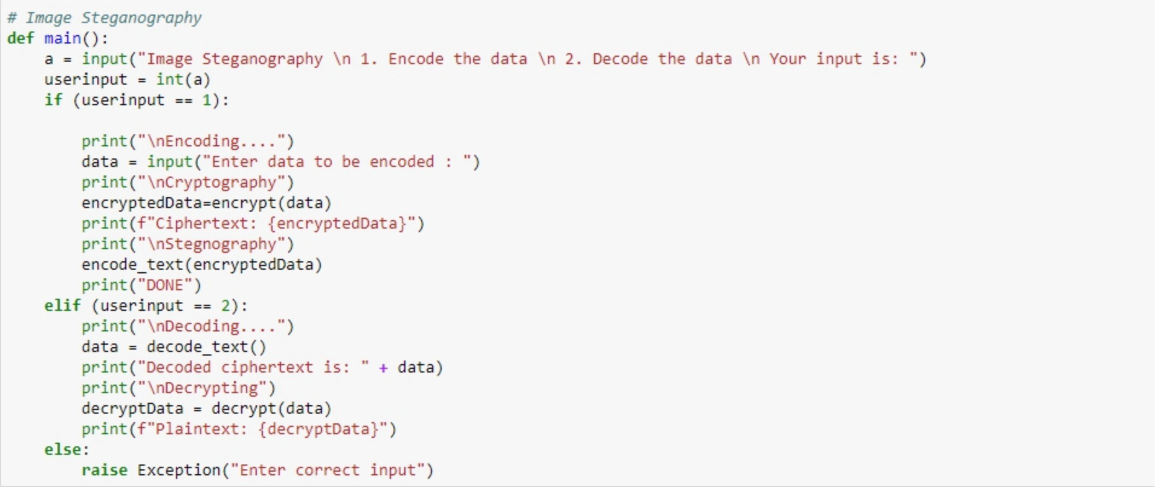


```
def Caesar(text,s):
    result = ""
    for i in range(len(text)):
        char = text[i]
        if(char==' '):
            result+=' '
        elif (char.isupper()):
            result += chr((ord(char) + s-65) % 26 + 65)
        else:
            result += chr((ord(char) + s - 97) % 26 + 97)
    return result
```

This function implements the common casear cypher which takes a string input along with shift value and then shifts the string characters by the value in shift variable.

This function then returns the modified string back to the calling function after completing this algorithm.

## Main function:



```
# Image Steganography
def main():
    a = input("Image Steganography \n 1. Encode the data \n 2. Decode the data \n Your input is: ")
    userinput = int(a)
    if (userinput == 1):

        print("\nEncoding....")
        data = input("Enter data to be encoded : ")
        print("\nCryptography")
        encryptedData=encrypt(data)
        print(f"Ciphertext: {encryptedData}")
        print("\nStegnography")
        encode_text(encryptedData)
        print("DONE")
    elif (userinput == 2):
        print("\nDecoding....")
        data = decode_text()
        print("Decoded ciphertext is: " + data)
        print("\nDecrypting")
        decryptData = decrypt(data)
        print(f"Plaintext: {decryptData}")
    else:
        raise Exception("Enter correct input")
```

This function is the main driver function for this project code.

This function starts by asking the user if they are receiver or sender.

Then for sender it asks for various inputs which include the selection of which algorithm to use for crypto (AES, casear cypher). Then the resulting data is save in the picture selected by the sender.

For receiver, this function asks as input for the image in which data is saved and then extracts the data, discards the image and sends this data for decryption.

Then this function returns the original data of sender to the reciever.

# Steganography_Report_Final1.pdf

*by*

**Submission date:** 24-Jun-2021 01:41PM (UTC+0530)
**Submission ID:** 1611482017
**File name:** Steganography_Report_Final1.pdf (2.92M)
**Word count:** 5432
**Character count:** 28504

# SECURING DATA USING STEGANOGRAPHY AND CRYPTOGRAPHY

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

**Computer Science and Engineering**

By

Ayush Singh (171335)

Akanksha Choudhary (171351)

Under the supervision of

Dr. Vivek Sehgal

to

Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh**

# Candidate's Declaration

I hereby declare that the work presented in this report entitled **Akanksha Choudhary** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2021 to June 2021 under the supervision of **Dr. Vivek Sehgal** (Computer Science).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature) Choudhary

Akanksha Choudhary, 171351

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Dr.Vivek Sehgal

Associate Professor

Computer Science

Dated: 18-05-2021

# Candidate's Declaration

I hereby declare that the work presented in this report entitled **Ayush Singh** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2021 to June 2021 under the supervision of **Dr. Vivek Sehgal** (Computer Science).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature) Ayush

Ayush Singh, 171335

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Dr.Vivek Sehgal

Associate Professor

Computer Science

Dated: 18-05-2021

# Acknowledgement

We would like to thank **Dr. Vivek Sehgal** for helping and guiding us throughout the semester in our project and supporting the concept.We have to put a lot of effort into this project, but this was not possible without the kind support and help of many people.So, we thank all of them sincerely. We are very grateful to our undertaking chief Dr. Vivek Sehgal for providing leadership and continuous [2] supervision as well as the necessary information about our project and for supporting the completion of this undertaking. We would like to give special thanks and appreciation to our friends and colleagues who have given us time and attention.

Thankyou.

Date : 18-05-2021

Ayush Singh(171335)

Akanksha Choudhary(171351)

# Table Of Content

**Content** **Page No.**

# List of Abbreviations

| | | |
|---|---|---|
| **AES** | - | Advanced Encryption Standard |
| **BMP** | - | Bitmap |
| **CFB** | - | Cipher Feedback |
| **GIF** | - | Graphic Interchange format |
| **JPEG** | - | Joint photographic experts group |
| **LSB** | - | Least Significant Bit |

# List of Figures

# List of Tables

# Abstract

One of the foremost important concern with the web users who are transmitting highly secure information over internet is secure data transfer, with the spread of computerized information round the globe through the internet, the safety of the data has raised a worry to the final population.

So, to beat this issue we includes the steganographic techniques. Using steganographic senders can hide their secure data within the kind of image, audio, and video files.

The receiver needs all the shares to recover the pictures with hidden message then the message to be decoded from the image.

The main objective behind this project is to transmit a message on a channel where another quite information is already being transmitted.The project is worried with the applying development to secure your data by converting your data in UTF-8 char codes

then embedding it in a picture so data is shielded from the intruders. The system makes the information double secure by embedding it in a picture and usage of RGB and UTF char codes. In this project user will choose a picture and convert the secret data into cipher text by using cryptography techniques and then embed the data into that image and at the receiver side the cipher text is decrypted from the image and by using decryption algorithms convert that cipher text into the original message.

# Chapter 01 : INTRODUCTION

## 1.1 Introduction

- While transfering a file from one point to a different through internet we'd like file secure concepts.

- This project helps us to send a file from one place to a different in a very secured manner.

- The software we employed in our project are:-
  - For backend-

    Jupyter Notebook

- Programming languages we used:-
  - Python

- By the algorithms which is predefined by the user. It is one in numerous information concealing techniques that transmit message over channel wherever another normal data is already being transmitted.

- Steganography means hiding the messages in such an easy way where nobody aside the user and receiver, will check the existence.

### 1.1.1 Encryption and Decryption

Encryption is said to be a process which transforms the initial information into an unrecognizable form.

At the encryption side we'll first take plain text and by using the encryption algorithm it will convert it into the cipher text.

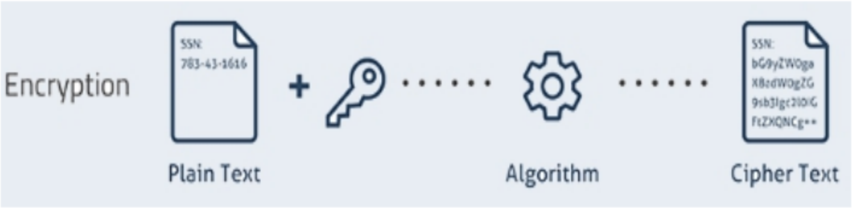


Fig.1.1 Encryption Process

Decryption said to be a process of converting encoded/encrypted data in an exceedingly form that's readable and understood by somebody's or a computer.

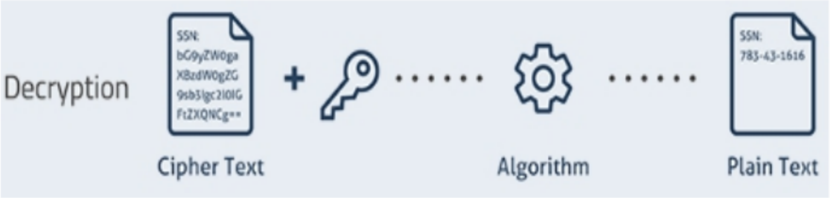


Fig.1.2 Decryption Process

### 1.1.2 Steganography and Cryptography

Cryptography may be a process which secures data using some algos in order so only that people for which the knowledge was meant will know it and process it.

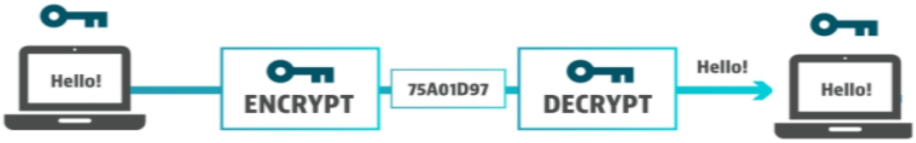


Fig.1.3 Cryptography

Steganography is one of many information hiding method, which aims to transmit the message over to the user where different kinds of data is send.

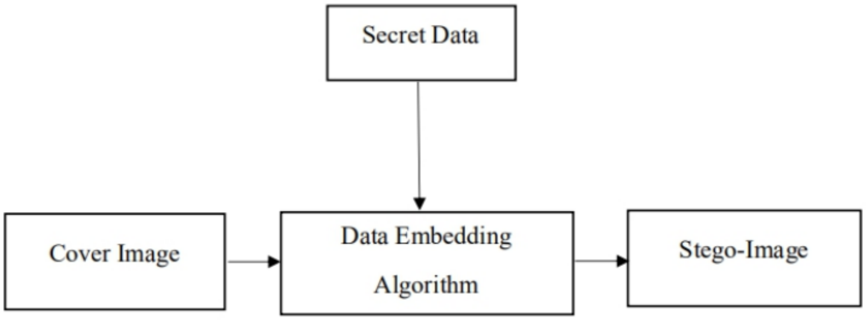


Fig.1.4 Steganography

The above figure shows that if a message is inserted in a picture through an algo, with the help of some other data or key. The result is stegnographic picture which send to other side where it’s analysed by the decryption algo.

Stego-image :- Stegnographic image is that which is the output of the process of embedding. Stego images have the messages hidden in either the values of pixels or in some optimally selected coefficients.

### 1.1.3 Some Modern Cryptography algorithms

- **Block Ciphers :**

  Block cipher consists for both algorithm that is encryption and decryption:

  1. First we give a key (K) to the sender and other text, and their product consists of cipher text block which is denoted by C.
  2. D is the decryption algo which is opposite of earlier algo in which cipher text is converted to original.

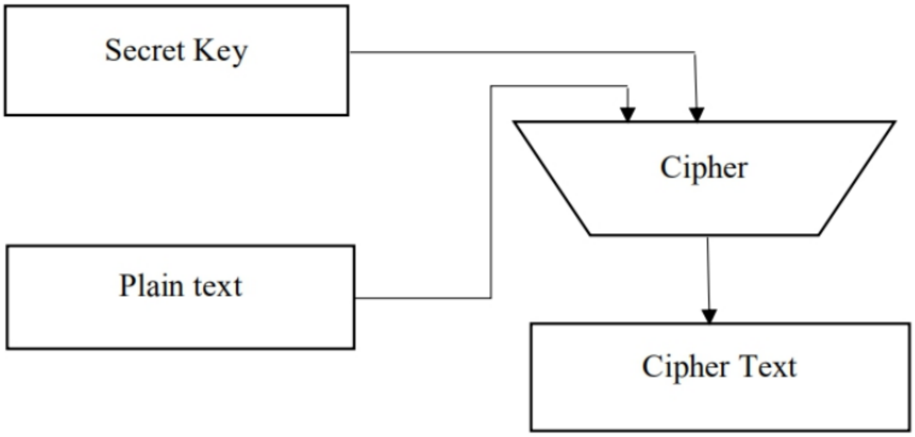


Fig.1.5 Block cipher diagram

- **Stream Ciphers :**

  It will treat pseudo random bits given from key, and therefore the plain text is converted by perform some operation with both the plain text and also the pseudo random bits. This type of cypher were sometimes ignored within the past.

  Below figure shows the asyn- and sync types.

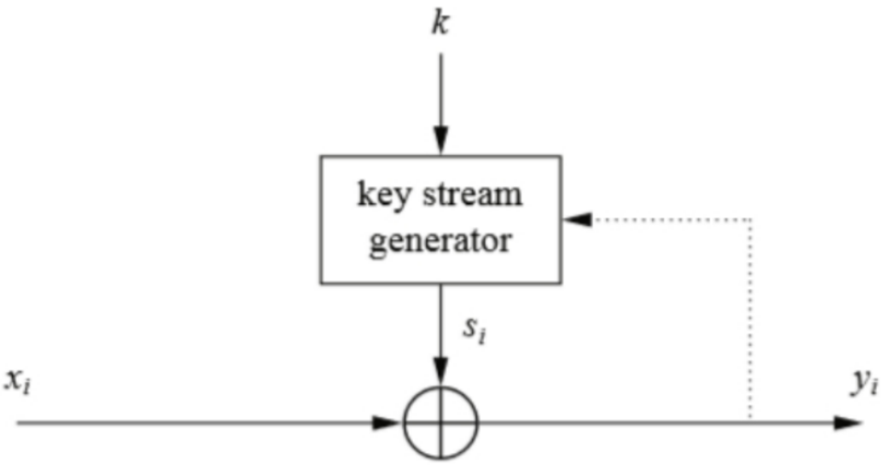


Fig.1.6 Asynchronous and synchronous types of stream ciphers

- **Hash Functions :**

  Earlier their is a function,which worked by maping by taking random input to a stucked output in an exceedingly process which is called compression, This process is not identical because the compression done in zip.

i. **Pre image collision resistance**: This type of property will generate when the input find a way to hash to give an output.

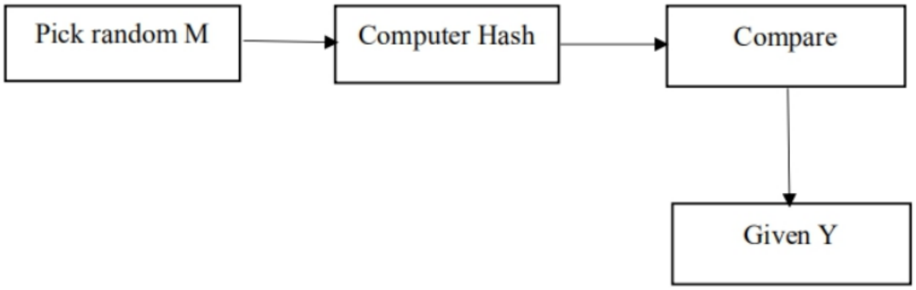


Fig.1.7 Preimage collision resistance

ii. **Second pre image collision resistance:** This is another way which is shown in below picture.

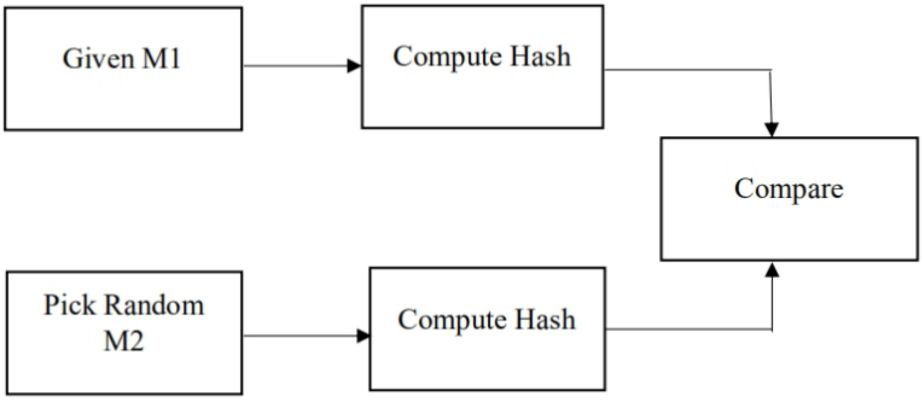


Fig.1.8 Second Preimage collision resistance

- **Digital Signature :**

Unlike cryptography, digital signatures didn’t exist before the invetion of computers. The requirement arose for digital sgnatures to be discussed, especlly within the business environments where multiple parties happen and every must attempt to keeping their declarations and/or proposals. The subject of unforgotable signatures was first discussed centuries ago, except those were handwritten signatures. The though behind digital signatures was first

A. **Digital Signature Requirements :**

The connection b/w which created the link between sign and encryption came into present with the “digitalization” era in which we were currently living and aalso become the witness of that.

B. **Digital Signature Principles :**

The below picture will the show the process of this method:-

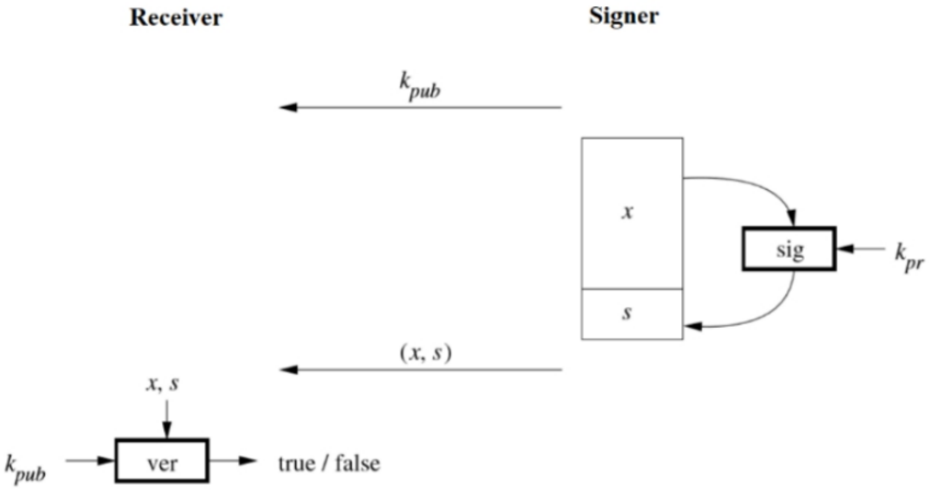


Fig.1.9 Digital Signature Principal (signing and verifying)

The function will require an external key so as order to link the signature to the sender who signed it, and also it become output of the verification function would be either "true" or "false". The output would be true in an exceedingly case during which the msg was signed through the private key thats linked with the opppositr key, the public verification key. Otherwise the output of the verification perform would be false.

### 1.1.4 Some Steganographic definitions :-

Below mentioned are some defintions which include :

**Cover medium :** his could be the another way to hide the knowledge.

**Embedded message :** It refers to a message which is secrectively enter inside a picture.

**Stego-medium :** It's often be often the last word piece of data that the can be seen.

### 1.1.5 Types of Steganography :-

- **Secret key Steganography :**

  Secret key steganography is outlined as a steganographic system that needs the exchange of a secret key (stego-key) before communication only the persons who know this secrete key can reverse the method using stego-key and skim the message. The benefit to Secret Key Steganography is whether or not it's intercepted, only parties who know the key can extract the key message.

- **Pure Steganography :**

  Pure steganography is defined as a steganographic system where there's no must exchange any password that's stego-key so as for the receiver to read the message. This method of Steganography is that the least secure means by which to speak secretly because the sender and receiver will bank solely upon the presumption that no alternative parties are responsive to this secret message.

### 1.1.6 Image Steganography :-

Because the name suggests it refers to the strategy of concealing information among all information. Picture chosen for this purpose is termed **cover-image** and therefore thepicture generated after this process is termed the **stego-image**.

Digital pictures those who seem on your laptop are jerky into pixels little dots with a selected color that along structure the image you ll be ready to see. For pictures steganographers code the message into the picture element lsb. This suggests that to the human eye the colour of the element diagram-matical by code to the pc doesn't change. The hidden msg was taken from the image provided something that know a) that there's a msg within the picture b) that you just use the identical steganographic program for decoding because the one want to hide the message.

Within this reason a text ciphertext alternative pictures or something which can be entered in a veryn exceedingly are going to be hidden in a picture. This process has come back quite so much in recent years with the event of quick graphical things and steganographic code is currently without delay on the market over Infobahn for everyday users.

**Concealment in digital images :**

Their are many processes by using which the data can be inserted in a picture, but we were using the below approach.

Common approaches include:

- **Least significant bit (LSB) insertion :**

  The right corner important bit insertion methodology is maybe the foremost renowned image steganography technique. It’s a typical straightforward approach to insert a data. Unfortunately, it’s extremely oberserved by attacks, such as image manipulation a straight forward conversion from a GIF format another format like jpeg.

  In below we’ll show how first 3 pixels will be 3 twenty four bit words below:

  (00100111 11101001 11001000)

  (00100111 11001000 11101001)

  (11001000 00100111 11101001)

  Letter A is having binary conversion is 10000011 which will bw ranging from highest left but:

  (0010011111101000 11001000)

  (00100110 11001000 11101000)

  (11001000 0010011111101001)

- **Masking and Filtering :**

To create a watermked picture, we add the lumnance of the masked area by fifteen percent. If we were to vary the luminance by a smaller percentage, the mask would be undetected by the human eye. Now we will use the watermarked image to cover plain text or encoded information.

Masking is more robust than LSB insertion with relevancy compression, cropping and some image processing. Masking techniques embed information in additional significant areas so the hidden message is more integral to the duvet image than simply hiding it within the "noise" level.

- **Algorithms and Transformations :**

To using redudant picture , you'll want to trade off the size of msg opposite to the robustess. An out sized message may be embedded just an occasion because it might occupy a far greater portion of the image area.

Apart from this another methods will perform some operations and scatter the hidden data throughut a picture. Scattering the msg makes it appear more like noise.

Scattring and encrypt will support to help against hidden msg to extract it from the picture but not against message destruction through image processing. A scattered message within he image's LSBs continues to be as prone to destruction from lossy compression and image processing, as may be a clear text message inserted within the LSBs.

### 1.1.7 Implementation of Steganography

Below picture will show that we will first a data then by using encrypting algo we'll convert that data into cypher text which will be used to insert in a picture which will be result in output image which is send over a channel to another side which is a receiver which extract that cypher text from picture and by detecting which algo will be used to convert data into cypher it will convert that data into original msg.

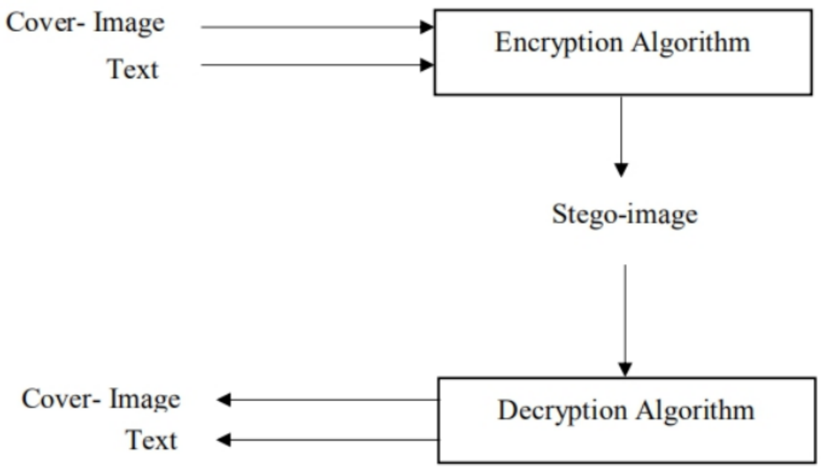


Fig.1.10 Process of Image Steganography

## 1.2 Problem Statement

Computer becomes more complicated and sophisticated, security becomes a major factor. Many securing data methods require several other methods. Secure transmissions are put in site to stop attacks like good spoofing and general data loss. Hence, in order to produce an improved mechanism, we are using RGB and UTF char codes during this report we will be using Steganography technique that's to embed our data into a picture in order that no outsider is attentive to the presence of knowledge.

Steganography has improved greatly in recent years, as many digital methods now allow hiding of data inside other information in new ways, & it could be importat in several conditions.

Steganography may be utilized in an outsized value of information formats in the present generation. Some of the data formats are .doc, .gif, .jpeg, .bmp, .wav, .txt and .mp3. Primarily due to their demand over the web and this method will use easy techniques. These formats are popular due to the relative ease by which redundant or noisy data are often far from them and replaced with a hidden message. Steganographic technologies are a awfully important a part of the longer term of Internet security and privacy on open systems like the Internet.

So, as to produce a steganography platform. We will convert our original data into cipher text and embed that cipher text into the picture by using LSB technique and at the receiver side we will decrypt the data from the picture and using the cryptography decryption algorithms convert that cipher text into the original message.

## 1.3 Objectives

The project has the subsequent objectives:

- To form an application which will be want to hide information.
- Authorization is needed to use the applying.
- The applying should be easy.
- The applying ought to encipher and decode the info.
- The applying provides enough security for your confidential information.

## 1.4 Methodology

Two pictures, key-image and encrypted-text-image, that are slightly different from each other, can be wont to calculate variations between their pixels which may be regenerate into utf char code which implies text.

### Data Hiding

Data concealment actual helpful information is scattered throughout the image and caps are full of random trash. Rgb that consists of three separate numbers are often accustomed hide knowledge well from any algorithmic program that searches for similarities between a pair of or additional footage that were geerated by a similar key the char code will be randomly distributed between the red, green and blue which will be added or subtracted from the original rgb. Therefore, if the same file is used to encode same data, still it will look different from previously made pictures and actual data pixels are undifferentiable from "trash" data, which also will change in a random way every time.

### 1.5 Organization

- In Chapter 1, Introduction and basic idea used in designing the project is mentioned. Objectives of the project, methodology exists in this chapter.

- In Chapter 2, Literature review is given, includes various research papers on Cryptography and Steganography, used to compare our results with the existing ones.

- In Chapter 3, System Development is discussed which includes software configuration and hardware configuration, front end and back end applications and their features.

- In Chapter 4, Performance Analysis is shown with the helps of screenshots of the application developed to ensure data security.

- In Chapter 5, Conclusion and Future scope of the project is mentioned.

# Chapter 02 : LITERATURE SURVEY

Encryption and secret writing a sort of hiding methods refers to the method of stealing data so the detector cannot sight the data.

In arithmetic applied science associated connected an formula is a good technique for finding a drag consits of several instructions. Algorithms are used widely for processing information and in plenty of different fields.Each algorithmic rule may be a list of well outlined directions for finishing a task.Starting from Associate in Nursing initial state the directions describe a computation that payoff through a well outlined series of ordered states eventually terminating in a very final ending state.The transition from one state to succeeding isn't essentially deterministic.Some algorithms called randomized algorithms incorporate randomness.

## 2.1 Cryptography

In this project, we will take some of the cryptographic techniques. First of all in our project their is a option given to the user which is chosen a algo to encrypt the data, so the user will encrypt the data according to them. We were using two methods i.e. Caesar cypher and AES encryption algorithm.

The first one will encrypt the data by shifting the alphabets where in case of second one it will having a advantage to choose a key. We will send the same key to other side so that if sender will choose AES method to encrypt the data then the other side will also choose the same to detect the original data.

## 2.2 Steganography

After converting our data to cypher text we use LSB technique to put our cypher text inside the picture. Basically we will take 500 x 500 x 3 size of our image and in this we will first convert our data into binary bits then each pixel is having rgb values we will take LSB bit of each pixek and replace it with our data bi.

Also we will add five hashes at the end so that when we send our data to receiver then its algo will retrieve that data from picture which also includes five hashes then algo will get to know that whenever five hashes will come it remove that hashes and remaining part will be our data.

Also at the end of the data their is a number written which shows that which algo would you to use to encrpyt the data then receiver's algo would get to konw and use the same algo and detect the original data.

**In this project we use following steganography :**

- Video Steganography
- Image Steganography
- Document Steganography
- Audio Steganography

Their are several types but we were using Image Steganorapthy for our project.

| METHODS | WEEK POINTS |
|---|---|
| Least Significant Bit | 1- Simple to extract<br>2- Compression can damage the data |
| Parity writing | Straightforward to extract and destroy |
| Echo concealing | Low inserting capability and security |
| Phase coding | Low capacity |
| Spread spectrum | 1-Requires more space.<br>2- Unprotected to time scale modification |
| Wavelet Domain | Retrive data might have lossy |

Table.2.1 Existing systems and their weak points

This paper [12] found out an image stego method that enhances the existing LSB substitution techniques techniques improve the protection level of hidden data and increase embedding capability of hidden data. Lossless compression technique are going to be utilised to compress secret data and hash operate is employed to hash the hidden information. Hash operate are going to be dead within the stego image and its values will be hold on in the hostimage for more checking throughout extraction process. The planned technique demonstrates considerably improvement in terms of knowledge security embedding capability and quality.

# Chapter 03 : SYSTEM DEVELOPMENT

## 3.1 Technology Used

### 3.1.1 Hardware Configuration

- Processor: 2.3 GHz Intel® Core™ or equivalent
- Memory: 2 GB

### 3.1.2 Software Configuration

- Operating System: Windows XP or higher
- Language: Python
- Framework: Jupyter Notebook

## 3.2 Jupyter Notebook

Jupyter Notebook and Jupyter Lab is an open source web based Integrated development environment for scientific computing and data science, but it can be used to quickly prototype almost any python based software.

## 3.3 Python

Python is one of the most popular programming language, which is high-level object oriented, which has dynamic semantics, and is interpreted. It also has some high-level built in data structures. It has dynamic binding and also typing. It is an extremely readable and intuitive language. With the huge number of useful libraries freely available for python, it is very fast to use and to prototype applications. Programmers also prefer python due to the increase productivity that comes by using it, because it is an interpreted language, which means that it does not have an extra step of compiling. Moreover, testing and debugging is very easy.

**Few benefits of python are:**

- It has vast community
- It has huge number of free libraries
- It is open source
- It is very secure
- It can work on any platform
- It is very readable and simple to understand
- It is simple to write
- Codes are generally concise
- Quick prototyping, due to interpreter based
- Python can work in procedural execution, oop as well as with functions.
- It can handle big data
- It can perform complex maths.

There are numerous uses of python, and programmers find newer ways to use python all the time. Few of the common uses of python are in Web development, Software dev, Data Science, Maths, Machine Learning, Artificial Intelligence, System Scripting.

## 3.4 Use Case Diagram

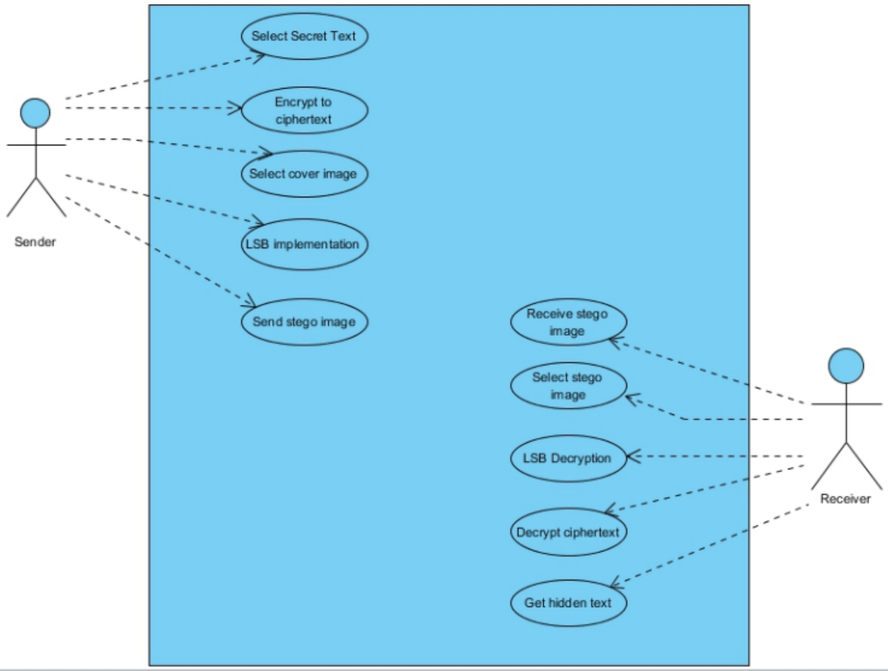


Fig.3.1 Use Case Diagram

### 3.5 Data Flow Diagram

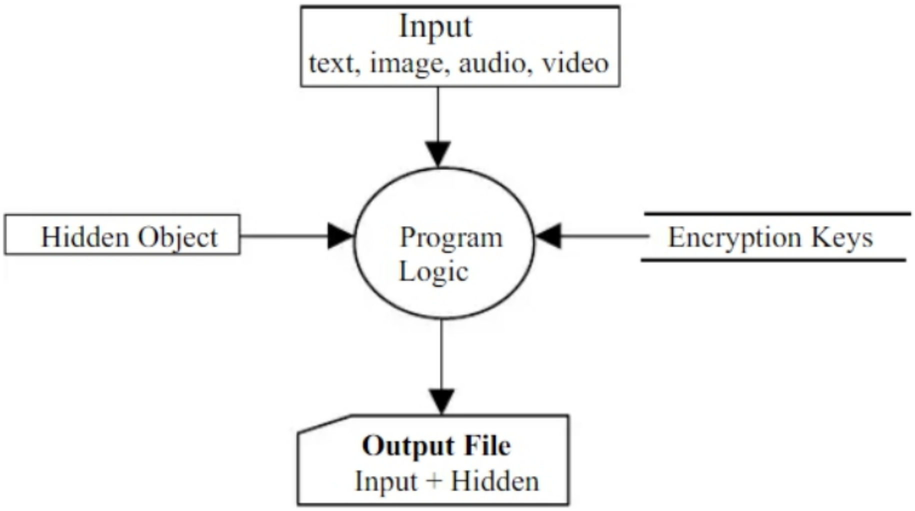


Fig.3.2 Data Flow Diagram

### 3.6 Comparison of Proposed and existing methods

Some Existing methods are not secure enough so here are some points in Table 3.3 which compares our proposed method with the existing ones.

| **Steganography** | **Existing Algorithms** |
|---|---|
| It provides double security to your data | Single level security is provided |
| Probability of detecting the hidden data is less | As, the data is shuffled on the basis of applied algorithm it can be cracked easily |
| RGB algorithm changes only least bit of the pixel which do not create any visible change in the image | Algorithm is applied to whole data so if encrypted data is visible one can easily find way to decrypt it |

Table.3.1 Proposed and Existing methods

## Chapter 04 : PERFORMANCE ANALYSIS

- **Home Page**

When we run the project, first window appears as shown in (Fig 4.1). In this window, we have two buttons:

**LOG-IN :** User can direct login into application.

**SIGN-UP :** New user can make their own account and can use the application.

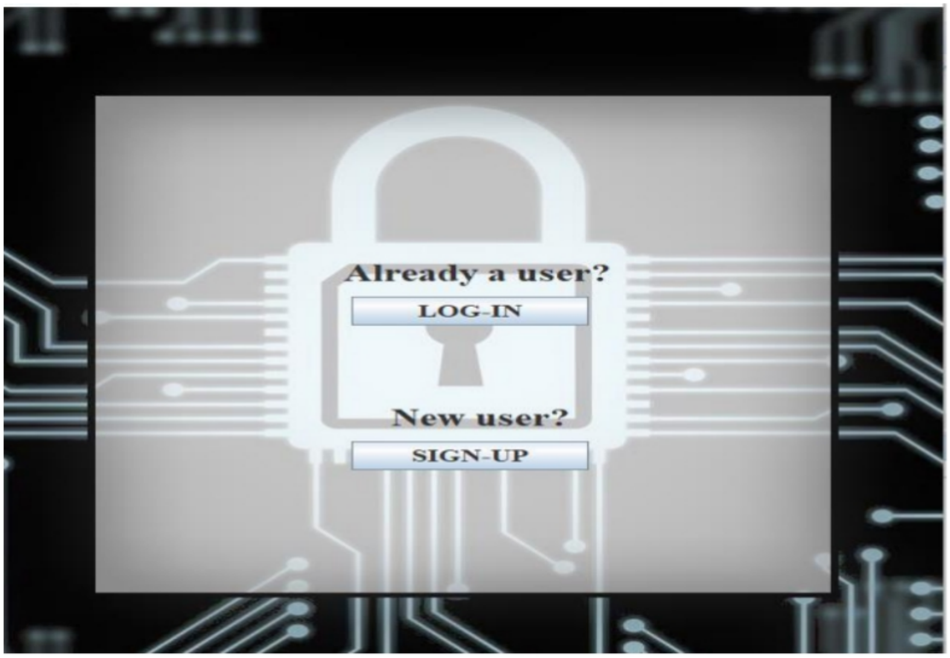


Fig.4.1 Application Home Page

- **Login Page**

When you click on the LOG-IN Button appears on the Window, it has 3 buttons:

**LOG-IN :** Opens the LOGIN page again.

**LOGIN :** User will go to the next Steg page.

**Back :** User will go the Home Page.

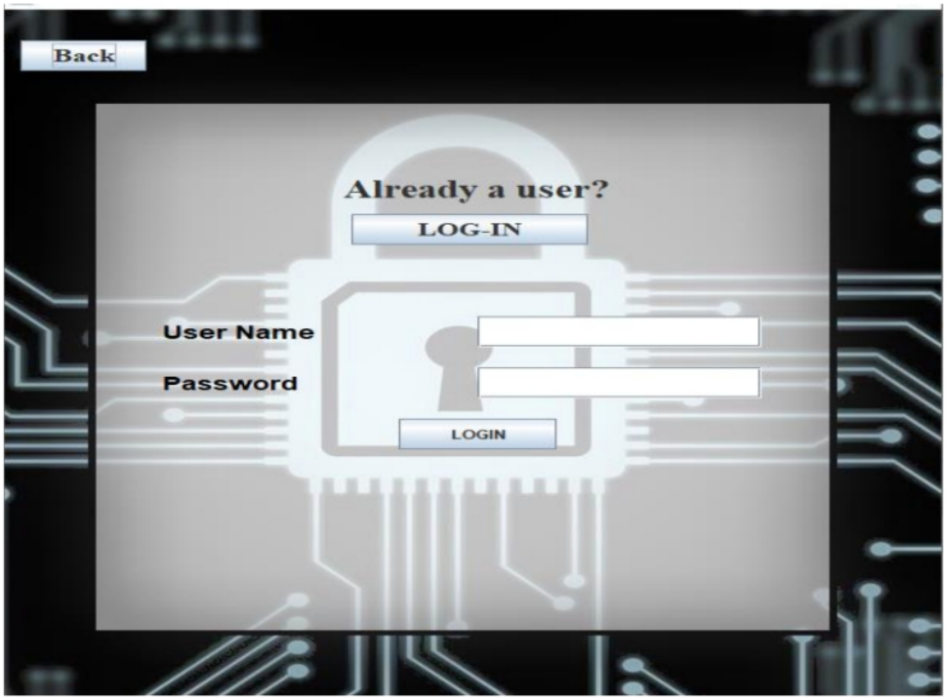


Fig.4.2 Login Page

- **Sign-Up Page**

When you click on the SIGN-UP Button appears on the Window, it has 3 buttons :

**SIGN-UP :** Opens the SIGN-UP page again.

**Back:** User will go the Home Page.

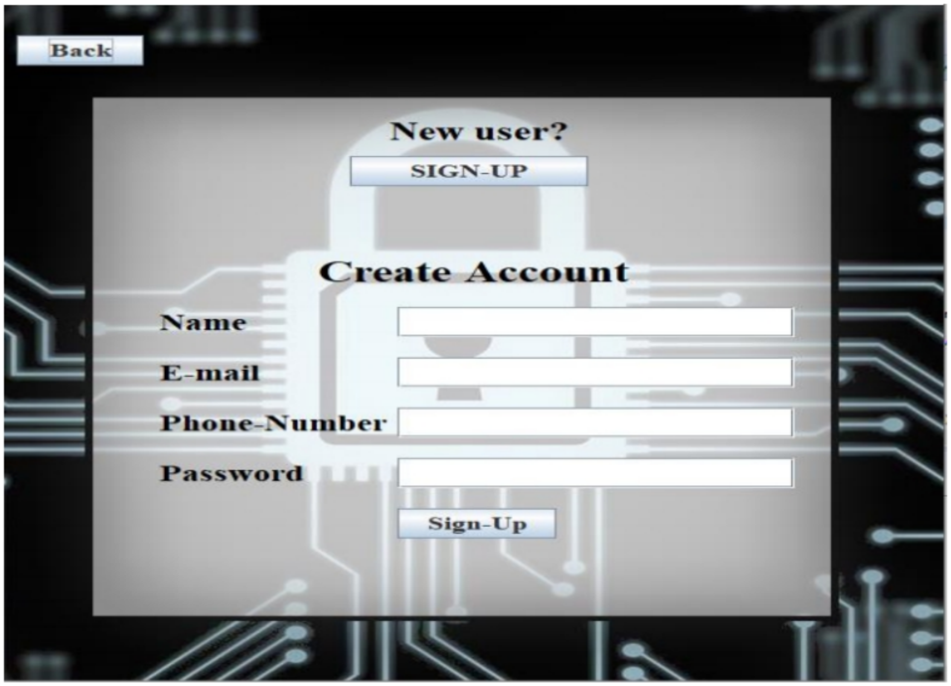


Fig.4.3 Sign-Up Page

- **Encryption & Decryption Choice**

Once user has Logged-In into the Application (Fig. 4.4) appears in the Window, it has two buttons, and a label with your User name:

**Encrypt:** User can encrypt his data.

**Decrypt:** User can decrypt his data.

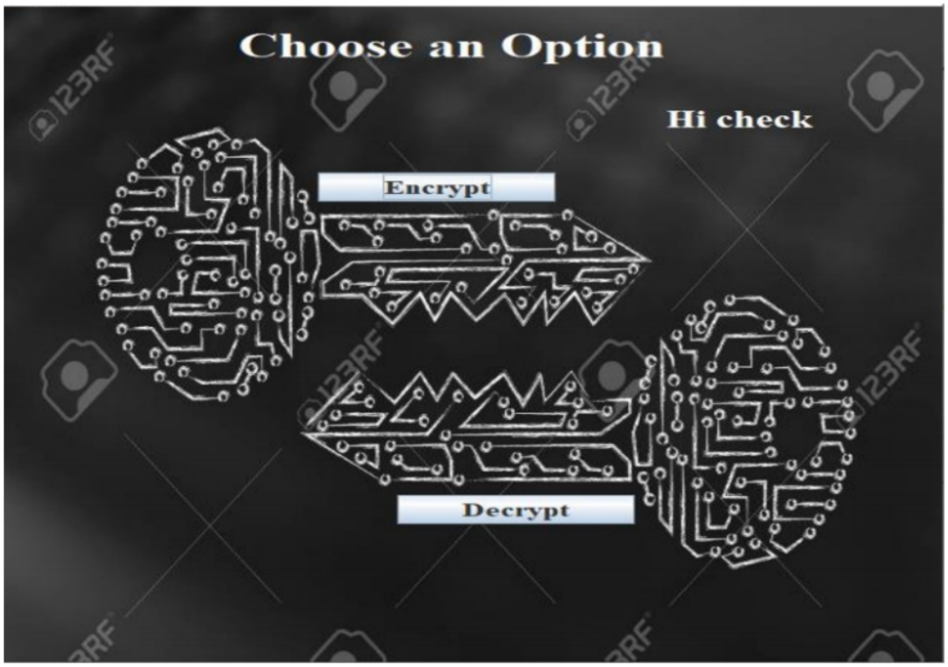
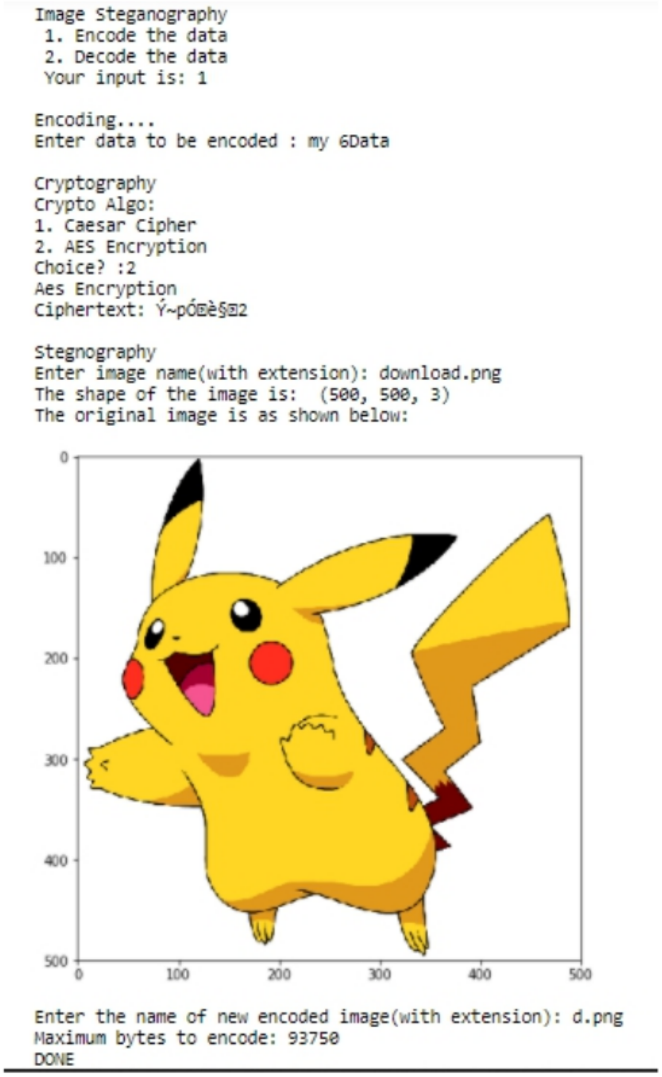


Fig.4.4 Encryption & Decryption Choice

- **Encryption**



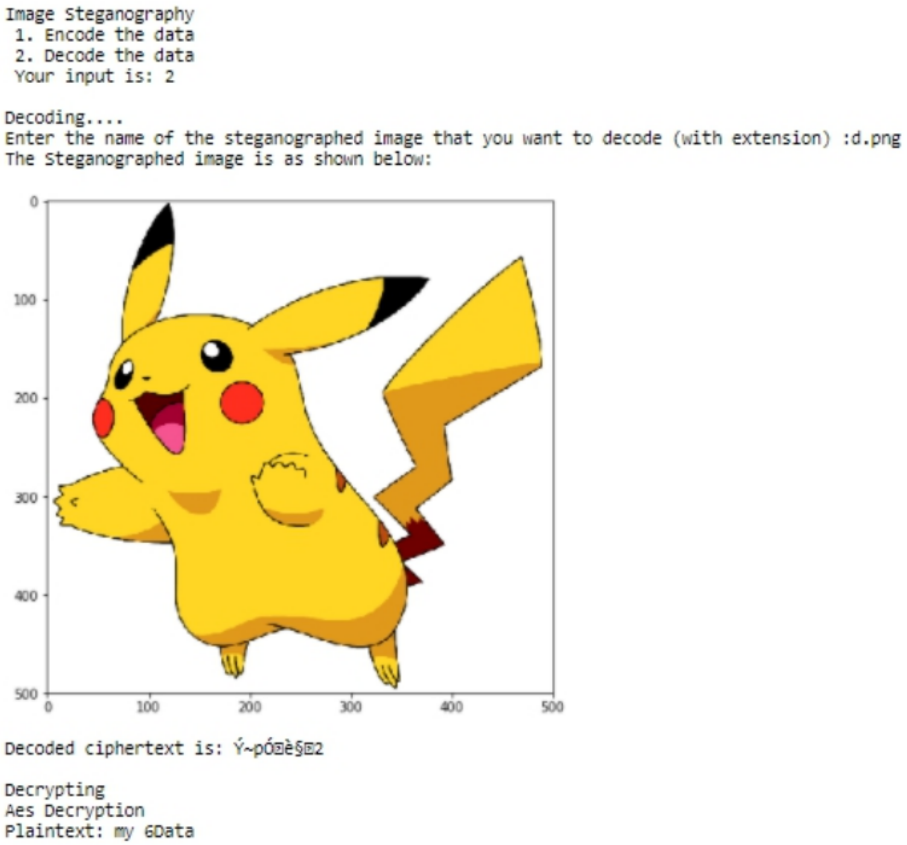
```
Image Steganography
 1. Encode the data
 2. Decode the data
 Your input is: 1

Encoding....
Enter data to be encoded : my 6Data

Cryptography
Crypto Algo:
1. Caesar Cipher
2. AES Encryption
Choice? :2
Aes Encryption
Ciphertext: Ý~pÓ⊠è§⊠2

Stegnography
Enter image name(with extension): download.png
The shape of the image is:  (500, 500, 3)
The original image is as shown below:
```

```
Enter the name of new encoded image(with extension): d.png
Maximum bytes to encode: 93750
DONE
```

Fig.4.5 Encrypted Image

Secret data is converted into the cipher text, then cipher text and image is input by the user and that text is encoded into the image and stego image is the output.

- **Decryption**



```
Image Steganography
 1. Encode the data
 2. Decode the data
 Your input is: 2

Decoding....
Enter the name of the steganographed image that you want to decode (with extension) :d.png
The Steganographed image is as shown below:
```

```
Decoded ciphertext is: Ý~pÓ⊡è§⊡2

Decrypting
Aes Decryption
Plaintext: my 6Data
```

Fig.4.6 Decryption

Stegano graphed image name is entered by the user and it will give the decrypted cipher text and by using decryption algorithm convert that cipher text into decrypted message.

## Chapter 05 : CONCLUSIONS

### 5.1 Conclusion

Recent advances in the field of Data Security have opened the way for the practical implementation of various types of Encryption and Steganography methods. After going through several paper, it was found that for implementing Data security there are various encryption algorithms which are not providing satisfactory security. So, this project is designed for implementing the steganography technique on a very basic RGB algorithm which was found to be more efficient than other basic algorithms.

### 5.2 Future Scope

**This project can be further upgraded in following areas:**

- To use more than one encryption algorithm and let give users option for selecting any one encryption algorithm for encrypting the file.
- To make application, use a public and private key to hide and extract the data.
- To make application which support video, audio embedding.
- To make application which support all formats for embedding.
- To make this application as a stand-alone application.

### 5.3 Applications

- Hiding a message with steganography strategies reduces the prospect of a message being detected.
- The benifit steganography has over cryptography techniques, by itself is that messages do not attract any suspicion to themselves.
- Difficult to detect, only if receiver has prior knowledge of presence of secret message.
- Can be used on any digital image or audio or video file with sufficient size.

## References

- Johnson, N. and Jajodia, S., “Exploring Steganography: Seeing the Unseen”,IEEE, Volume: 31 , Feb. 1998, pp.26-34.
- C. Kurak, J. McHugh, "A Cautionary Note On Image Downgrading", *Proc. IEEE Eighth Ann. Computer Security Applications Conf.*, pp. 153-159, 1992.
- TanmyBhaowmik, PramathaNathBasu, “On Embedding of text in Audio – A case of Steganography”, International Conference on Recent Trends in Information, Telecommunication and computing, IEEE 2010.
- W. D. A. M. E. HELLMAN, "New directions in cryptography," IEEE Transactions on Information Theory, Vols. IT-22, no. 6, pp. 644-654, 1976.
- S. J. Lincke and A. Hollan, "Network Security: Focus on Security, Skills, and Stability," in 37th ASEE/IEEE Frontiers in Education Conference, Milwaukee, 2007.
- H. Orman, "Recent Parables in Cryptography," IEEE Internet Computing, vol. 18, no. 1, pp. 82-86, 2014.
- R. GENNARO, "IEEE Security & Privacy," IEEE Security & Privacy, vol. 4, no. 2, p. 64 - 67, 2006.
- B. Preneel, "Cryptography and Information Security in the Post-Snowden Era," in IEEE/ACM 1st International Workshop on TEchnical and LEgal aspects of data privacy and SEcurity, Florence, 2015.
- Johnson, N. and Jajodia, S., “Exploring Steganography: Seeing the Unseen”,IEEE, Volume: 31 , Feb. 1998, pp.26-34.
- C. Kurak, J. McHugh, "A Cautionary Note On Image Downgrading", Proc. IEEE Eighth Ann. Computer Security Applications Conf., pp. 153-159, 1992.
- B. Preneel, "Cryptography and Information Security in the Post-Snowden Era," in IEEE/ACM 1st International Workshop on Technical and LEgal aspects of data privacy and security, Florence, 2015.

# Appendices

- **Algorithm**

**Algorithm for embedding data inside image :**

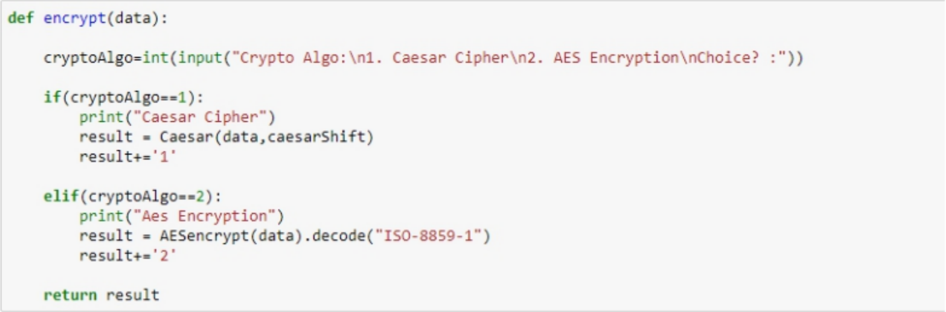
```
Convert Secret_Message to Cipher_text;
Input:    Cover_Image, Cipher_text;
    Convert Secret_Message to bits;
    Imbed Bit_Secret_Message to Cover_Image pixels;
Output: Stego_image;
```

**Algorithm for extracting data from stego image :**

```
Input : Stego_image;
        Decode all Bit_Secret_Message form Stego_image by each pixel;
        Convert Bit_Secret_Message to Cipher_text_Message;
        Convert cipher_text_Message to Secret_Message;
Output : Secret_Message;
```

- **Code**

**Convert data into cipher text :**

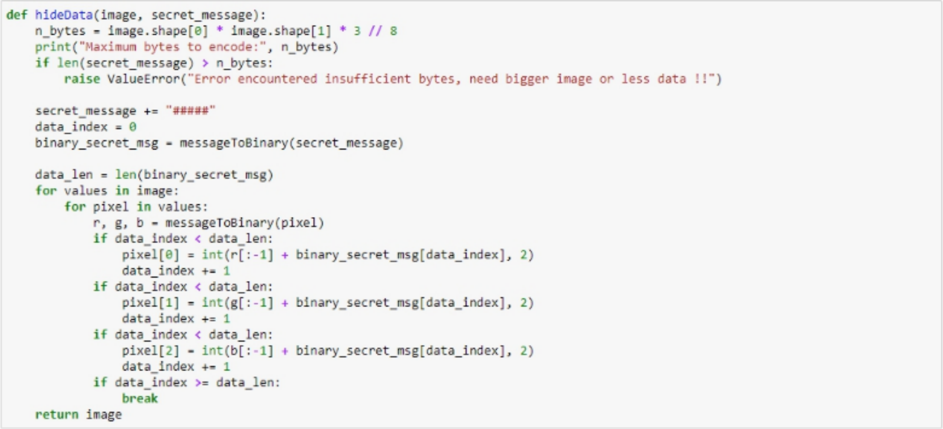


```
def encrypt(data):

    cryptoAlgo=int(input("Crypto Algo:\n1. Caesar Cipher\n2. AES Encryption\nChoice? :"))

    if(cryptoAlgo==1):
        print("Caesar Cipher")
        result = Caesar(data,caesarShift)
        result+='1'

    elif(cryptoAlgo==2):
        print("Aes Encryption")
        result = AESencrypt(data).decode("ISO-8859-1")
        result+='2'

    return result
```

In this piece of code we'll do cryptography. Basically, by asking the user that which algo they want to use to convert the original text into the cypher text.

Also by doing we assign that part to the result variable and at the end we will add some number which will help the receiver program to find out which algo was used to make cypher, so that they can recover the original msg sent by the sender.

### Hide cipher text into image :

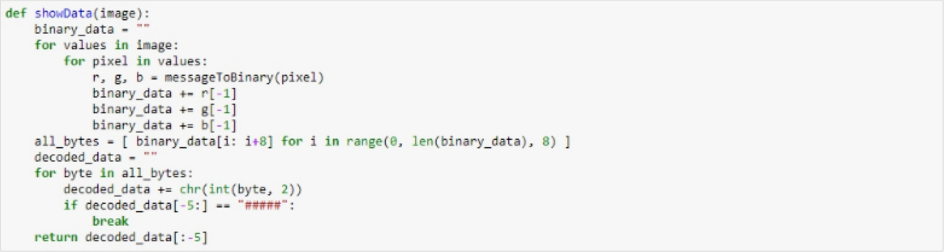


```
def hideData(image, secret_message):
    n_bytes = image.shape[0] * image.shape[1] * 3 // 8
    print("Maximum bytes to encode:", n_bytes)
    if len(secret_message) > n_bytes:
        raise ValueError("Error encountered insufficient bytes, need bigger image or less data !!")

    secret_message += "#####"
    data_index = 0
    binary_secret_msg = messageToBinary(secret_message)

    data_len = len(binary_secret_msg)
    for values in image:
        for pixel in values:
            r, g, b = messageToBinary(pixel)
            if data_index < data_len:
                pixel[0] = int(r[:-1] + binary_secret_msg[data_index], 2)
                data_index += 1
            if data_index < data_len:
                pixel[1] = int(g[:-1] + binary_secret_msg[data_index], 2)
                data_index += 1
            if data_index < data_len:
                pixel[2] = int(b[:-1] + binary_secret_msg[data_index], 2)
                data_index += 1
            if data_index >= data_len:
                break
    return image
```

This function reads this picture, prints its shape, converts the msg into binary and then adds "#####" at the end of msg, so that the receiver can figure out when the msg is terminated.

This function then converts the data into the picture red, green and blue pixel values.

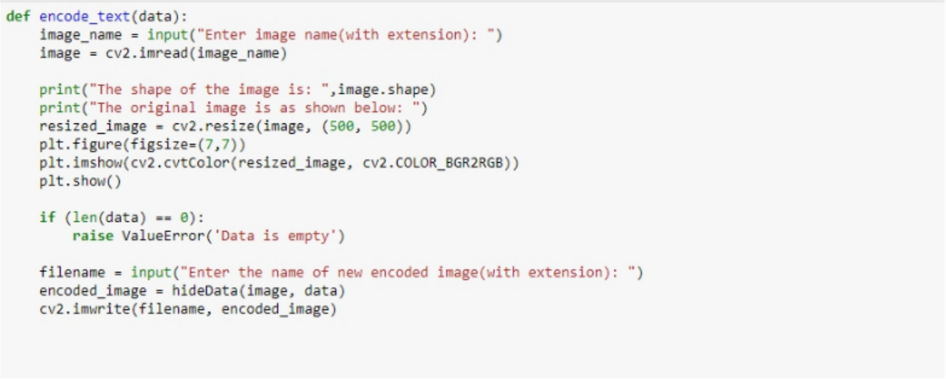
### Extract hidden cipher text from image :



```
def showData(image):
    binary_data = ""
    for values in image:
        for pixel in values:
            r, g, b = messageToBinary(pixel)
            binary_data += r[-1]
            binary_data += g[-1]
            binary_data += b[-1]
    all_bytes = [ binary_data[i: i+8] for i in range(0, len(binary_data), 8) ]
    decoded_data = ""
    for byte in all_bytes:
        decoded_data += chr(int(byte, 2))
        if decoded_data[-5:] == "#####":
            break
    return decoded_data[:-5]
```

The above code extracts the data from picture red, green and blue pixel values and stops reading the data when “#####” are encountered and then removes the hashes and returns the resulting data.

**Input and pre-process data before encoding :**

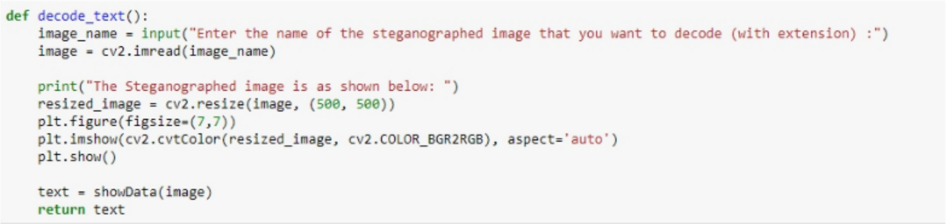


```
def encode_text(data):
    image_name = input("Enter image name(with extension): ")
    image = cv2.imread(image_name)

    print("The shape of the image is: ",image.shape)
    print("The original image is as shown below: ")
    resized_image = cv2.resize(image, (500, 500))
    plt.figure(figsize=(7,7))
    plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
    plt.show()

    if (len(data) == 0):
        raise ValueError('Data is empty')

    filename = input("Enter the name of new encoded image(with extension): ")
    encoded_image = hideData(image, data)
    cv2.imwrite(filename, encoded_image)
```

This function reads the picture from intrenal storage in same folder and then processes it and sends it to ‘hide data function’.

This function then saves this resulting picture from RAM back to intrenal storage in same folder as a new file.

### Input image to decode, and display hidden message :

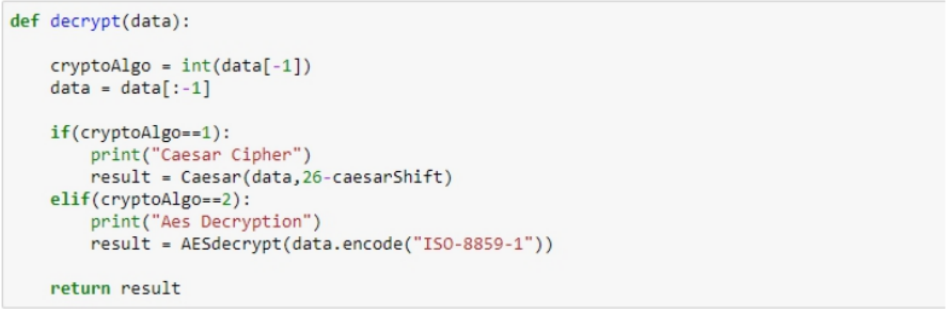


```python
def decode_text():
    image_name = input("Enter the name of the steganographed image that you want to decode (with extension) :")
    image = cv2.imread(image_name)

    print("The Steganographed image is as shown below: ")
    resized_image = cv2.resize(image, (500, 500))
    plt.figure(figsize=(7,7))
    plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB), aspect='auto')
    plt.show()

    text = showData(image)
    return text
```

This function reads the resulting picture from intrenal storage in same folder and then processes it and sends it to ‘show data function’.

This function then returns the resulting data to the other side and discards the picture in RAM.

### Convert hidden message(cipher text) into original data :

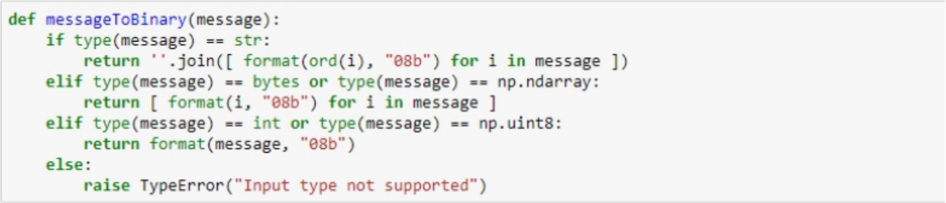


```python
def decrypt(data):

    cryptoAlgo = int(data[-1])
    data = data[:-1]

    if(cryptoAlgo==1):
        print("Caesar Cipher")
        result = Caesar(data,26-caesarShift)
    elif(cryptoAlgo==2):
        print("Aes Decryption")
        result = AESdecrypt(data.encode("ISO-8859-1"))

    return result
```

This function receives cypher text along with a last bit which shows that what crypto algo was used.

This function then uses that algorithm to convert the cypher into original value and then it returns to the other side.

**Converting input to binary:**

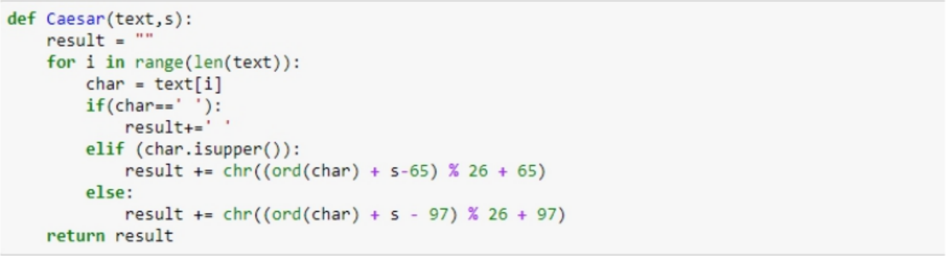


```
def messageToBinary(message):
    if type(message) == str:
        return ''.join([ format(ord(i), "08b") for i in message ])
    elif type(message) == bytes or type(message) == np.ndarray:
        return [ format(i, "08b") for i in message ]
    elif type(message) == int or type(message) == np.uint8:
        return format(message, "08b")
    else:
        raise TypeError("Input type not supported")
```

This function receives an input from multiple finctions and detects the type of input data and then converts it into binary data and returns back this binary data to the calling function.

This function also supports string type data, byte typr data, array type data and integer typr data.

**Casear Cipher:**

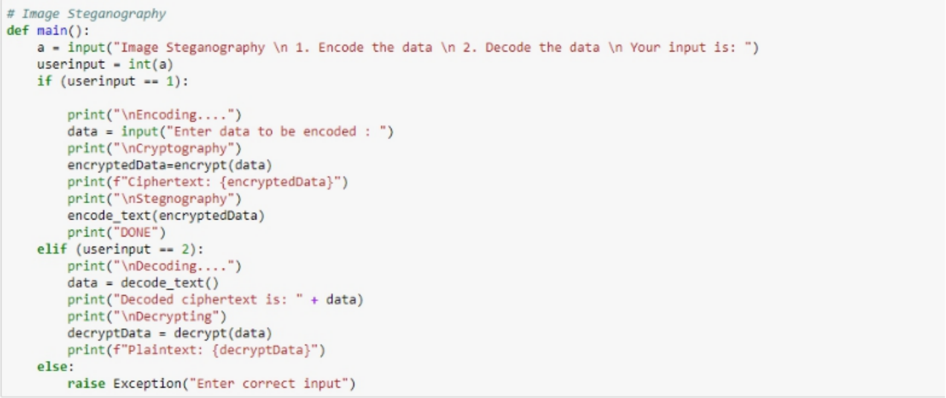


```
def Caesar(text,s):
    result = ""
    for i in range(len(text)):
        char = text[i]
        if(char==' '):
            result+=' '
        elif (char.isupper()):
            result += chr((ord(char) + s-65) % 26 + 65)
        else:
            result += chr((ord(char) + s - 97) % 26 + 97)
    return result
```

This function implements the common casear cypher which takes a string input along with shift value and then shifts the string characters by the value in shift variable.

This function then returns the modified string back to the calling function after completing this algorithm.

### Main function:



```python
# Image Steganography
def main():
    a = input("Image Steganography \n 1. Encode the data \n 2. Decode the data \n Your input is: ")
    userinput = int(a)
    if (userinput == 1):

        print("\nEncoding....")
        data = input("Enter data to be encoded : ")
        print("\nCryptography")
        encryptedData=encrypt(data)
        print(f"Ciphertext: {encryptedData}")
        print("\nStegnography")
        encode_text(encryptedData)
        print("DONE")
    elif (userinput == 2):
        print("\nDecoding....")
        data = decode_text()
        print("Decoded ciphertext is: " + data)
        print("\nDecrypting")
        decryptData = decrypt(data)
        print(f"Plaintext: {decryptData}")
    else:
        raise Exception("Enter correct input")
```

This function is the main driver function for this project code.

This function starts by asking the user if they are receiver or sender.

Then for sender it asks for various inputs which include the selection of which algorithm to use for crypto (AES, casear cypher). Then the resulting data is save in the picture selected by the sender.

For receiver, this function asks as input for the image in which data is saved and then extracts the data, discards the image and sends this data for decryption.

Then this function returns the original data of sender to the reciever.

Steganography_Report_Final1.pdf

ORIGINALITY REPORT

| 13% | 12% | 4% | 9% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.ukessays.com<br>Internet Source | 4% |
| 2 | www.ir.juit.ac.in:8080<br>Internet Source | 2% |
| 3 | Submitted to Jaypee University of Information Technology<br>Student Paper | 2% |
| 4 | Abdalbasit Mohammed Qadir, Nurhayat Varol. "A Review Paper on Cryptography", 2019 7th International Symposium on Digital Forensics and Security (ISDFS), 2019<br>Publication | 1% |
| 5 | patents.justia.com<br>Internet Source | 1% |
| 6 | Submitted to Manav Rachna International University<br>Student Paper | 1% |
| 7 | Submitted to nith<br>Student Paper | <1% |

| | | |
|---|---|---|
| 8 | M.S. Borella, J.P. Jue, D. Banerjee, B. Ramamurthy, B. Mukherjee. "Optical components for WDM lightwave networks", Proceedings of the IEEE, 1997<br>Publication | <1 % |
| 9 | Submitted to International Health Sciences University<br>Student Paper | <1 % |
| 10 | Submitted to Texas A & M University, Kingville<br>Student Paper | <1 % |
| 11 | www.coursehero.com<br>Internet Source | <1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 14 words |
| Exclude bibliography | On | | |

# JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

## PLAGIARISM VERIFICATION REPORT

**Date: 16-06-2021**

**Type of Document (Tick):** **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

**Name:** Ayush Singh________________ **Department:** Computer Science___ **Enrolment No:** 171335____

**Contact No.** +91-9752428599________________**E-mail.** ayush14singh@gmail.com_________________

**Name of the Supervisor:** Dr. Vivek Sehgal____________________________________________________

**Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):** SECURING DATA USING STEGANOGRAPHY AND CRYPTOGRAPHY_____________________________________________________

## UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages = 49
- Total No. of Preliminary pages = 8
- Total No. of pages accommodate bibliography/references = 1

Ayush

**(Signature of Student)**

## FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ___13___(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Vivek Sehgal

**(Signature of Guide/Supervisor)** **Signature of HOD**

## FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

| **Copy Received on** | **Excluded** | **Similarity Index (%)** | **Generated Plagiarism Report Details** (Title, Abstract & Chapters) | |
|---|---|---|---|---|
| 24-Jun-2021 | • All Preliminary Pages<br>• Bibliography/Images/Quotes<br>• 14 Words String | 13 | Word Counts | 5432 |
| **Report Generated on** | | | Character Counts | 28504 |
| 24-Jun-2021 | | **Submission ID** | Total Pages Scanned | 49 |
| | | 1611482017 | File Size | 19.2 KB |

**Checked by**
**Name & Signature** **Librarian**

………………………………………………………………………………………………………………………………………………

**Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com**

# JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

## PLAGIARISM VERIFICATION REPORT

**Date: 16-06-2021**

**Type of Document (Tick):** **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

**Name:** Akanksha Choudhary____________ **Department:** Computer Science___ **Enrolment No:** 171351__

**Contact No.** +91-8219282294__________________**E-mail.** akankshachoudhary94505@gmail.com________

**Name of the Supervisor:** Dr. Vivek Sehgal____________________________________________________________

**Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):** SECURING DATA USING STEGANOGRAPHY AND CRYPTOGRAPHY____________________________________________________________

## UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages = 49
- Total No. of Preliminary pages = 8
- Total No. of pages accommodate bibliography/references = 1

Choudhary

**(Signature of Student)**

## FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ___13___ (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Vivek Sehgal

**(Signature of Guide/Supervisor)** **Signature of HOD**

## FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

| **Copy Received on** | **Excluded** | **Similarity Index (%)** | **Generated Plagiarism Report Details** (Title, Abstract & Chapters) | |
|---|---|---|---|---|
| 24-Jun-2021 | • All Preliminary Pages<br>• Bibliography/Images/Quotes<br>• 14 Words String | 13 | Word Counts | 5432 |
| **Report Generated on** | | | Character Counts | 28504 |
| 24-Jun-2021 | | **Submission ID** | Total Pages Scanned | 49 |
| | | 1611482017 | File Size | 19.2 KB |

**Checked by**
**Name & Signature** **Librarian**

……………………………………………………………………………………………………………………………………………

**Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com**