STUDENT DETAILS

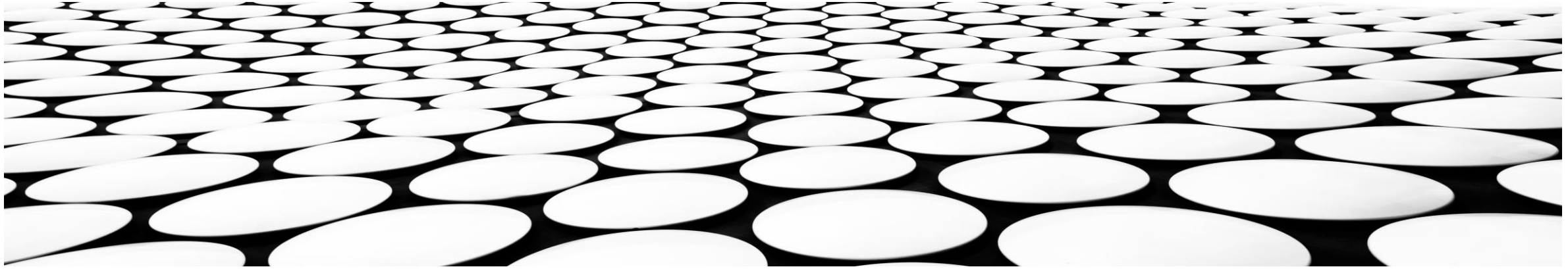
NAME: NARAYANA HEMA KIRAN

E-MAIL: NARAYANAHEMAKIRAN@GMAIL.COM

COLLEGE NAME: SRI VASAVI ENGINEERING COLLEGE

COLLEGE STATE: ANDHRA PRADESH

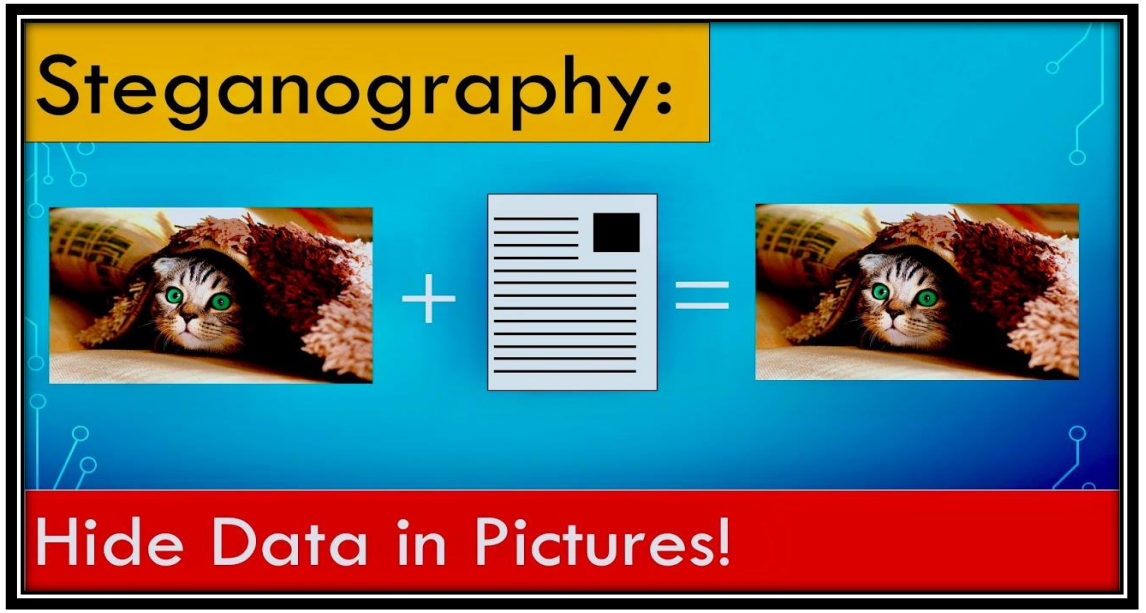
INTERNSHIP DOMAIN: CYBER SECURITY



HIDING A TEXT INSIDE AN IMAGE USING STEGANOGRAPHY

- An officer in danger requires a method to send a secret text message without alerting nearby criminals to his higher officials. Traditional communication methods are easily intercepted or monitored, so, he his necessitating a covert technique to embed and transmit critical information securely to his higher officials.

AGENDA



- The agenda of this project to send a secret text to higher officials using the steganography concept and RGB technique, ensuring that criminals or others cannot identify the image or its hidden content. Thos project will involve encoding the secret text into the least significant bits of the image's pixels, making the alterations imperceptible to the human eye. Additionally, a secure key-based XOR operation will be used to enhance the security of the hidden message. Finally, the project will include a robust mechanism for decoding and retrieving the hidden message, ensuring that only authorized personnel with the correct key can access the concealed information.

PROJECT OVERVIEW

- This project implements a steganography technique to securely hide and reveal secret within an image using the RGB color mechanism and a key based XOR operation for encryption. Steganography allows for concealing messages within non-secret text or data, ensuring secure communication by embedding sensitive information within an image in a way that is imperceptible to unauthorized viewers.
- **Image Encoding:** Secret text is hidden within the image's pixel values using the LSB method, with additional security provided by XOR the text characters with a user-provided security key.
- **Pixel Manipulation:** The encoded text is distributed across the image's pixels, maintaining the visual integrity of the image while embedding the hidden message.
- **Text Decoding:** The project includes functionality to decrypt and retrieve the hidden text from the image using the correct security key, ensuring that only authorized users can access the information.
- This project demonstrates a practical application of steganography for secure communication, embedding and retrieving secret messages within images in a secure and efficient manner.

WHO ARE THE END USERS OF THIS PROJECT?

- The primary end users for this steganography project include government and military personnel secure communication, corporate executives protecting proprietary data, and journalists sharing sensitive information discreetly.
- IT and cyber security teams can integrate this technique to enhance organizational data security.
- Additionally, the person who wants to send any secret message other person in a hidden format then this project will helpful for them.

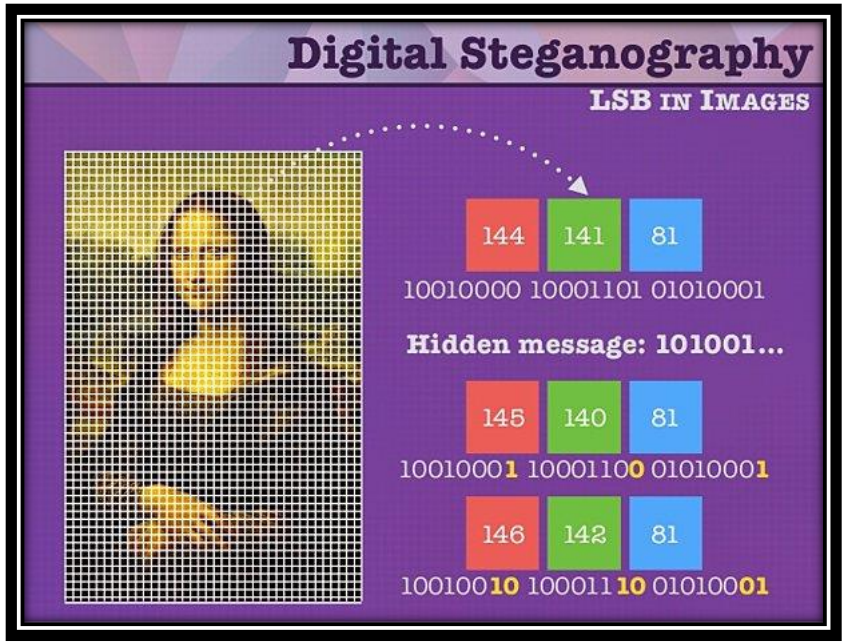
YOUR SOLUTION AND ITS VALUE PROPOSITION

.

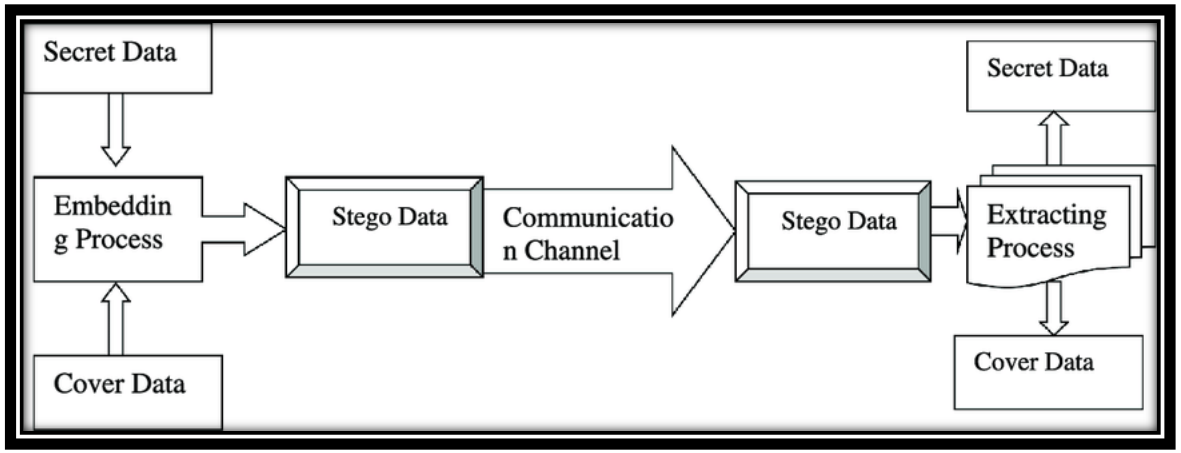
- Steganography is the basic concept to hide the data inside other data.
- In this project RGB Mechanism is used for pixel manipulation.
- XOR operation is used for encryption and decryption of the test inside the image.
- The project reads an image and hides the secret text within the pixel values using the least significant bits (LSB) method.
- For security purpose that means to avoid unauthorized users taking the advantage of the message, a secret key used to hide and unhide the data.
- Finally, By using this project we can hide the data inside an image using secret key and for unhide the message secret is used.

HOW DID YOU CUSTOMIZE THE PROJECT AND MAKE IT YOUR OWN

- By using problem statement this project is created to hide a text inside an image using RGB, LSB method is used for imperceptible to the human eye. And here XOR operation is used for hide and unhide the image.
- This project is customized only to hide a text inside an image using above conditions and can only unhide the image who has secret key others are not able to unhide the image.



MODELLING



- **Step 1:** importing some libraries like cv2 and os for accessing relevant concept into code.
- **Step 2:** After converting the text into their ascii values then that ascii values are stored in variable.
- **Step 3:** Read the image from it's path and hiding the image using XOR operation , RGB mechanism
- **Step 4:** A secret is created to avoid unauthorized users.
- **Step 5:** To unhide the image user wants to enter the secret key.
- **Step 6:** Finally, user can able to see the secret message.

RESULTS

```
print(result)
```

```
[ ]
```

```
# Install necessary packages
!pip install opencv-python-headless

import cv2
import numpy as np
import os
from google.colab import files

# Create dictionaries for encoding and decoding
d = {chr(i): i for i in range(255)}
c = {i: chr(i) for i in range(255)}

# Upload image file
uploaded = files.upload()

# Read the image
for fn in uploaded.keys():
    x = cv2.imdecode(np.fromstring(uploaded[fn], np.uint8), cv2.IMREAD_COLOR)

i, j, k = x.shape
print(i, j, k)

# Input security key and text to hide
key = input("\nEnter key to edit (Security Key): ")
text = input("\nEnter text to hide:")
```



```
    if key == key1:
        for i in range(l):
            decrypt += c[x[n, m, z] ^ d[key[k1]]]
            z = (z + 1) % 3
            if z == 0:
                m = (m + 1) % j
                if m == 0:
                    n = (n + 1) % i
            k1 = (k1 + 1) % len(key)
        print("The Secret Message is:", decrypt)
    else:
        print("Check your key!!!!")
else:
    print("Don't Want To Unhide The Text, Ok Then Bye!!!!")
```

```
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (from opencv-python-headless) (1.25.2)
Choose Files  image.png
• image.png(image/png) - 58624 bytes, last modified: 7/5/2024 - 100% done
<ipython-input-3-3ad6c0f85b0b>:18: DeprecationWarning: The binary mode of fromstring is deprecated, as it behaves surprisingly on unicode inputs. Use frombuffer instead
  x = cv2.imdecode(np.fromstring(uploaded[fn], np.uint8), cv2.IMREAD_COLOR)
Saving image.png to image (1).png
163 166 3

Enter key to edit (Security Key): 3

Enter text to hide:secuty
Data Hiding in Image completed successfully.

Enter 1 To Unhide The Text: 1

Enter Secret key To Unhide The Text: 0
Check your key!!!!
```

INPUT AND OUTPUT

Input image

Encrypted image

LINK:-

- https://github.com/narayanahemakiran/cybersecurity