CAPSTONE PROJECT

SECURE DATA HIDING IN IMAGES USING STEGANOGRAPHY

Presented By:
Student Name : Chaitanya Anil Soni
College Name & Department : Sandip University

edunet
foundation

OUTLINE

- Problem Statement
- Technology used
- Wow factor
- End users
- Result
- Conclusion
- Git-hub Link
- Future scope

PROBLEM STATEMENT

In an increasingly digital world, the need for secure communication has never been greater. Traditional encryption methods, while effective, often raise suspicion and attract unwanted attention. This project aims to develop an advanced steganography method that optimizes data capacity, enhances resistance to steganalysis, and preserves the quality of the carrier file, ensuring the secure and undetectable transmission of sensitive information.

TECHNOLOGY USED

- Libraries:

‘cv2’ is OpenCV’s python library which hosts an extensive collection of over 2500 optimized algorithms – both classic and state-of-the-art computer vision and machine learning. Created to realize the full potential of computer vision, cv2 in Python facilitates easy integration of real-world data to help machines perceive visuals like a human.

- Platforms: Python

WOW FACTORS

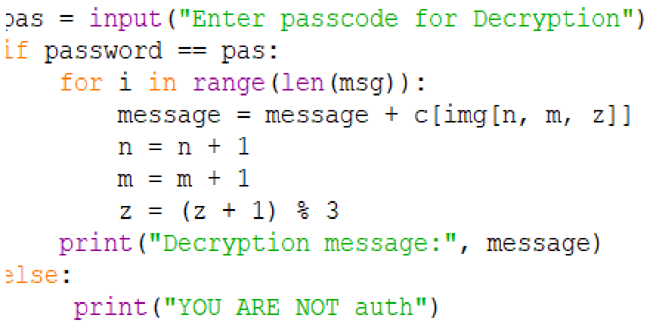
Steganography is like the secret agent of the digital world. It stands out because it hides information in plain sight, making it nearly invisible to the untrained eye. Unlike encryption, which scrambles data into unreadable gibberish, steganography embeds hidden messages within ordinary files, like images, audio, or text. This makes it a powerful tool for covert communication and data protection. Imagine sending a secret message hidden within a seemingly innocent vacation photo—pretty cool.

END USERS

- Who are the end users

RESULTS

Decryption code

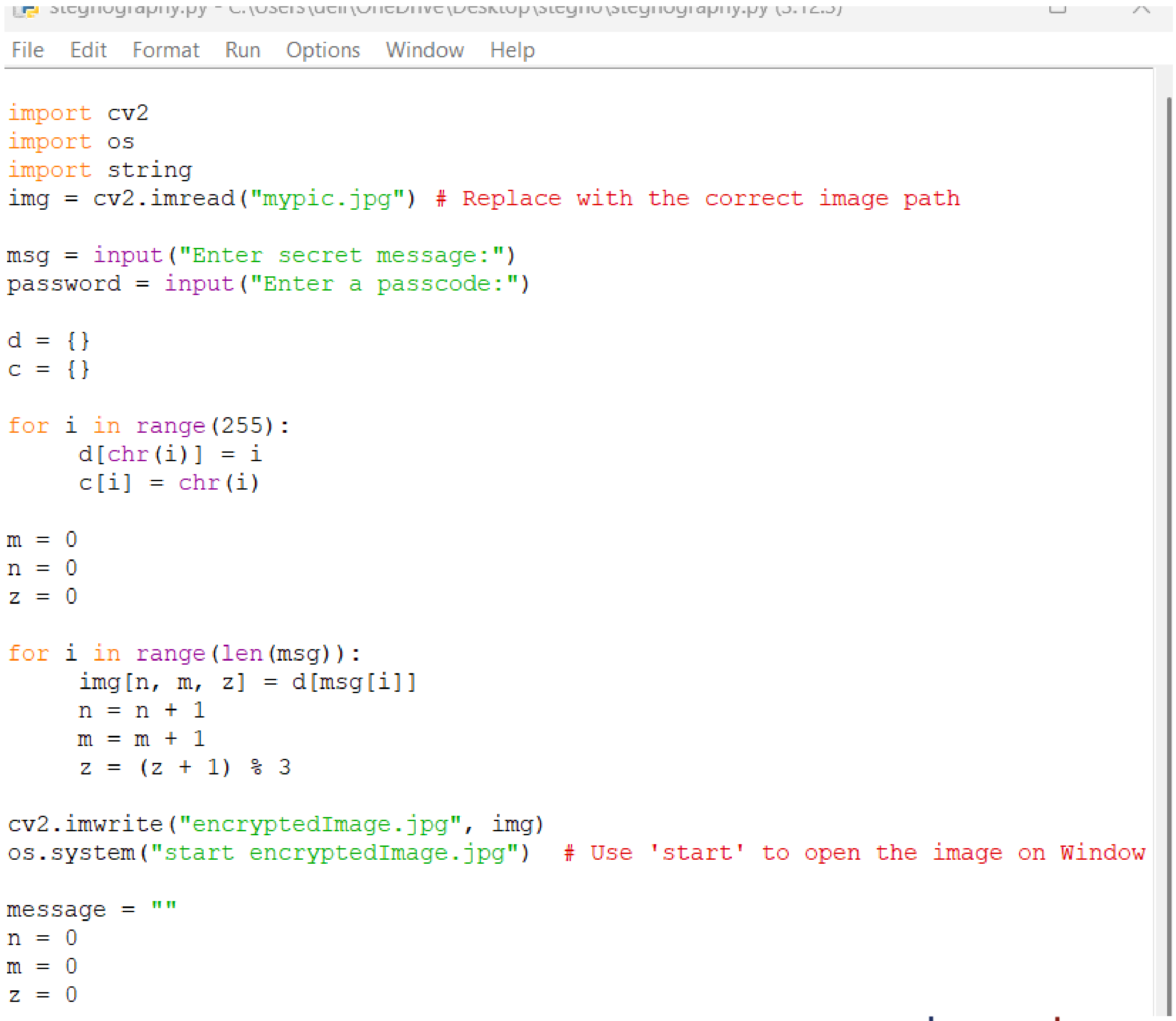


```
pas = input("Enter passcode for Decryption")
if password == pas:
    for i in range(len(msg)):
        message = message + c[img[n, m, z]]
        n = n + 1
        m = m + 1
        z = (z + 1) % 3
    print("Decryption message:", message)
else:
     print("YOU ARE NOT auth")
```

Original Image

Encrypted image

ecryption code


stegnography.py - C:\Users\dell\OneDrive\Desktop\stegno\stegnography.py (3.12.5)
File
Edit
Format
Run
Options
Window
Help

```
import cv2
import os
import string
img = cv2.imread("mypic.jpg") # Replace with the correct image path

msg = input("Enter secret message:")
password = input("Enter a passcode:")

d = {}
c = {}

for i in range(255):
     d[chr(i)] = i
     c[i] = chr(i)

m = 0
n = 0
z = 0

for i in range(len(msg)):
     img[n, m, z] = d[msg[i]]
     n = n + 1
     m = m + 1
     z = (z + 1) % 3

cv2.imwrite("encryptedImage.jpg", img)
os.system("start encryptedImage.jpg")  # Use 'start' to open the image on Window

message = ""
n = 0
m = 0
z = 0
```

CONCLUSION

- **Techniques Used:**
- **Least Significant Bit (LSB) Substitution**: This common method involves changing the least significant bits of the cover medium to encode the hidden message. For example, in a digital image, the color values of individual pixels are subtly altered.
- **Spread Spectrum**: This technique spreads the hidden message across a wide range of frequencies, making it harder to detect and remove.
- **Masking and Filtering**: Similar to watermarking, this method hides the message by blending it into the cover medium, leveraging its inherent properties.
- **Advantages:**
- **Invisibility**: Unlike encryption, which makes data unreadable, steganography hides information in plain sight, making it less likely to attract attention.
- **Dual Functionality**: The cover medium retains its original purpose, whether it's an image, audio file, or document, while also serving as a vessel for the hidden message.

GITHUB LINK

- https://github.com/ChaitanyaSoni010/Steganoraphy

THANK YOU