CAPSTONE PROJECT

SECURE DATA HIDING IN IMAGE USING STEGANOGRAPHY

Presented By:
Student Name : DINESH SRIDHARAN
College Name & Department : ANNAI VAILANKANNI ARTS AND SCIENCE , INFORMATION TECHNOLOGY

edunet foundation

OUTLINE

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

PROBLEM STATEMENT

- **Introduction:**
  - Cyberattacks and data breaches are on the rise.
  - Sensitive information is vulnerable to interception and hacking.

- **Challenge:**
  - Need for innovative methods to securely transmit data.
  - Traditional security methods (e.g., encryption) are often vulnerable to attacks.

- **Solution:**
  - **Steganography:** Hiding data within images to prevent unauthorized access.

TECHNOLOGY USED

- **Steganography Techniques:**
  - **Least Significant Bit (LSB):** Embedding data in the least significant bits of an image.
  - **Discrete Cosine Transform (DCT):** Hiding data within the image's frequency domain.

- **Image Processing Tools:**
  - OpenCV (Python) or PIL (Python Imaging Library) for image manipulation.

- **Encryption (Optional):**
  - AES Encryption for securing data before embedding.

- **Programming Languages:**
  - Python (preferred for simplicity and versatility).

WOW FACTORS

- **Invisible Data**: Data hidden in an image without visible changes.
- **Two Layers of Security**: Combination of steganography and encryption.
- **Real-World Use Cases**: Secure communication, digital watermarking, and private data storage.
- **Increased Privacy**: Provides a more secure alternative to conventional data transmission methods.

END USERS

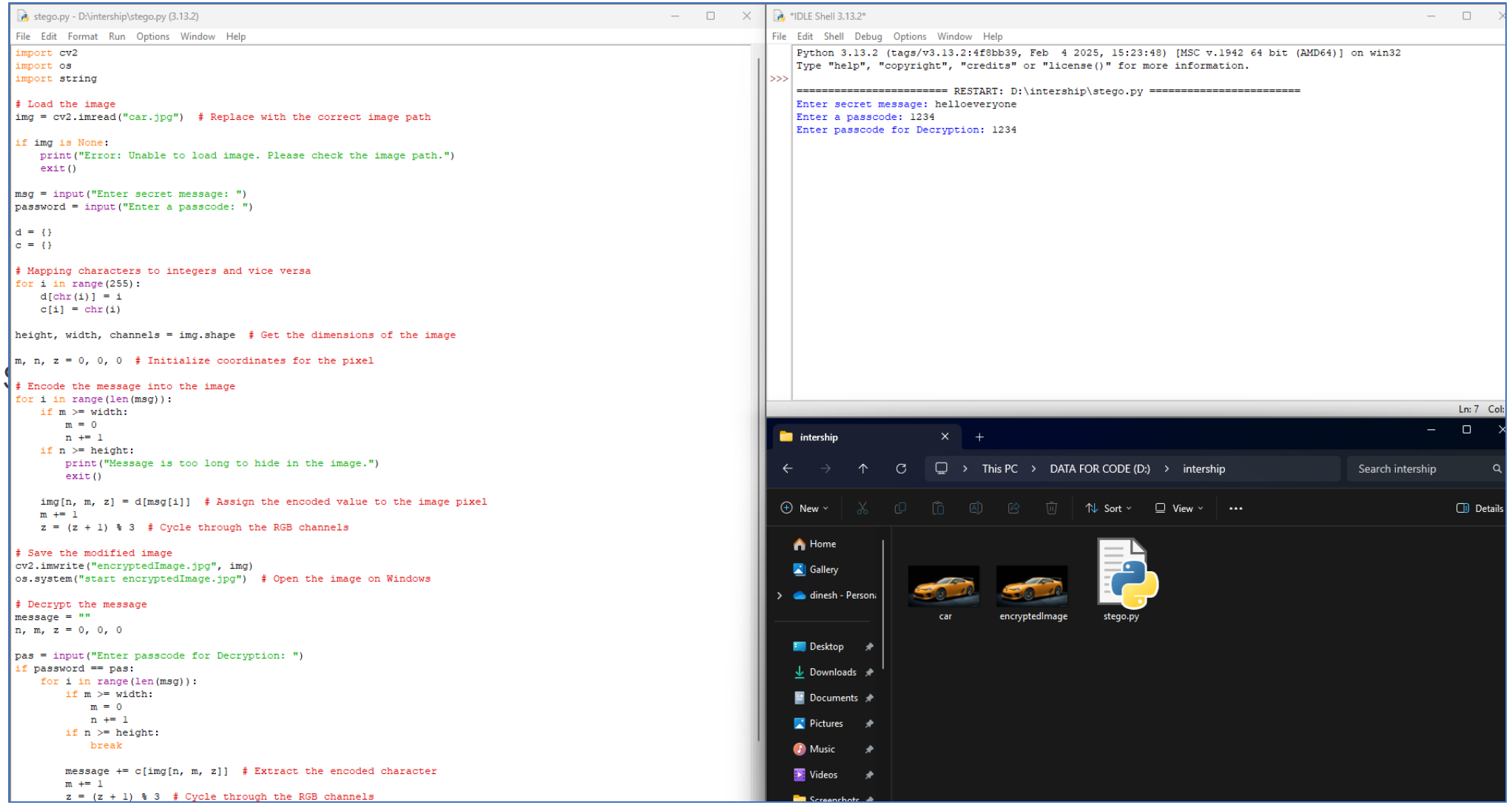
- **Corporate Sector:**
  - Securely send sensitive business documents.

- **Government Agencies:**
  - Protect confidential communications, especially in intelligence operations.

- **Individuals:**
  - Secure personal data (photos, documents) shared online.

- **Cybersecurity Professionals:**
  - Use in secure data exchanges and to detect hidden threats.

RESULTS


stego.py - D:\intership\stego.py (3.13.2)
File Edit Format Run Options Window Help
import cv2
import os
import string
# Load the image
img = cv2.imread("car.jpg") # Replace with the correct image path
if img is None:
print("Error: Unable to load image. Please check the image path.")
exit()
msg = input("Enter secret message: ")
password = input("Enter a passcode: ")
d = {}
c = {}
# Mapping characters to integers and vice versa
for i in range(255):
d[chr(i)] = i
c[i] = chr(i)
height, width, channels = img.shape # Get the dimensions of the image
m, n, z = 0, 0, 0 # Initialize coordinates for the pixel
# Encode the message into the image
for i in range(len(msg)):
if m >= width:
m = 0
n += 1
if n >= height:
print("Message is too long to hide in the image.")
exit()
img[n, m, z] = d[msg[i]] # Assign the encoded value to the image pixel
m += 1
z = (z + 1) % 3 # Cycle through the RGB channels
# Save the modified image
cv2.imwrite("encryptedImage.jpg", img)
os.system("start encryptedImage.jpg") # Open the image on Windows
# Decrypt the message
message = ""
n, m, z = 0, 0, 0
pas = input("Enter passcode for Decryption: ")
if password == pas:
for i in range(len(msg)):
if m >= width:
m = 0
n += 1
if n >= height:
break
message += c[img[n, m, z]] # Extract the encoded character
m += 1
z = (z + 1) % 3 # Cycle through the RGB channels
IDLE Shell 3.13.2
File Edit Shell Debug Options Window Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\intership\stego.py ==========
Enter secret message: helloeveryone
Enter a passcode: 1234
Enter passcode for Decryption: 1234
Ln: 7 Col:
intership
This PC > DATA FOR CODE (D:) > intership
Search intership
New
Sort
View
Details
Home
Gallery
dinesh - Persona
Desktop
Downloads
Documents
Pictures
Music
Videos
car
encryptedImage
stego.py

CONCLUSION

- **Effective Security Solution**: Steganography provides a unique and effective method to protect data.
- **Combining Encryption and Steganography**: Enhances security and privacy.
- **Practical Use**: Offers real-world applicability for secure data transmission.
- **Scalability**: Adaptable for various domains such as secure messaging, digital watermarks, and file protection.

GITHUB LINK

- HTTPS://GITHUB.COM/DINZS01/MY_PROJECT_01.GIT

FUTURE SCOPE(OPTIONAL)

- **Advanced Algorithms:**
  - Development of more robust algorithms that withstand attacks.

- **AI Integration:**
  - Use of AI to detect steganography or improve its detection.

- **Real-Time Applications:**
  - Implementations in secure communication apps or file-sharing platforms.

- **Multimedia Data:**
  - Hiding complex data like audio and video in addition to images.

THANK YOU

edunet foundation