CAPSTONE PROJECT

SECURE DATA HIDING IN IMAGE USING STEGANOGRAPHY

Presented By: Ponna Gunadeesh Reddy
Student Name : Ponna Gunadeesh Reddy
College Name & Department : Saveetha School Of Engineering,
Computer Science and Engineering

edunet
foundation

OUTLINE

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

PROBLEM STATEMENT

- With the increasing need for secure communication, traditional encryption techniques alone are not enough to ensure data confidentiality.
- Sensitive information can be intercepted and decrypted if detected.
- This project aims to provide an advanced and secure method of data hiding using image steganography, ensuring that the hidden message remains undetectable while maintaining the quality of the image.
- In highly sensitive fields like journalism, military operations, and corporate espionage protection, covert communication plays a crucial role. Simply encrypting data is not enough; it must also remain undetectable.
- Image steganography provides a unique solution by embedding secret messages within images in such a way that they appear unchanged to the human eye and most digital analysis tools.
- However, existing steganographic techniques often suffer from weaknesses such as noticeable distortions in image quality or susceptibility to detection by steganalysis tools.

TECHNOLOGY USED

Technology Used

- Integrated Development Environment (IDE):IDLE (Python's Built-in IDE) – A lightweight, beginner-friendly environment for writing, debugging, and running Python scripts.

- **Operating System:**
  Windows 10/11 (64-bit) – Compatible with Python and supports all required libraries and dependencies**.**

- **Programming Language:** Python – Chosen for its simplicity, efficiency, and vast library support for image processing and encryption.

- **Libraries:**
  1. **OpenCV** (pip install opencv-python) – Used for image processing, loading, modifying, and saving stego images.

     Command : pip install opencv-python

WOW FACTORS

- **Dual Security Approach:** Combines steganography with AES encryption, making data extraction extremely difficult without the correct decryption key.
- **Adaptive Steganography:** Dynamically adjusts data embedding to maintain image quality and reduce suspicion.
- **Multi-layer Encoding:** Implements different encoding strategies like frequency domain techniques (DCT-based hiding) for increased security.
- **Noise-Resistant Extraction:** Ensures that even minor image distortions (compression, resizing) do not easily corrupt the hidden data.
- **Real-Time Analysis:** A built-in verification tool to detect steganographic artifacts and ensure imperceptibility.

END USERS

•**Government & Intelligence Agencies** - Secure communication without detection.

•**Corporate Sector** - Secure transmission of confidential data and trade secrets.

•**Journalists & Activists** - Protect sensitive information in oppressive regimes.

•**Cybersecurity Professionals** - Implement and test new security measures.

•**General Users** - Privacy-focused individuals who want to securely share messages.

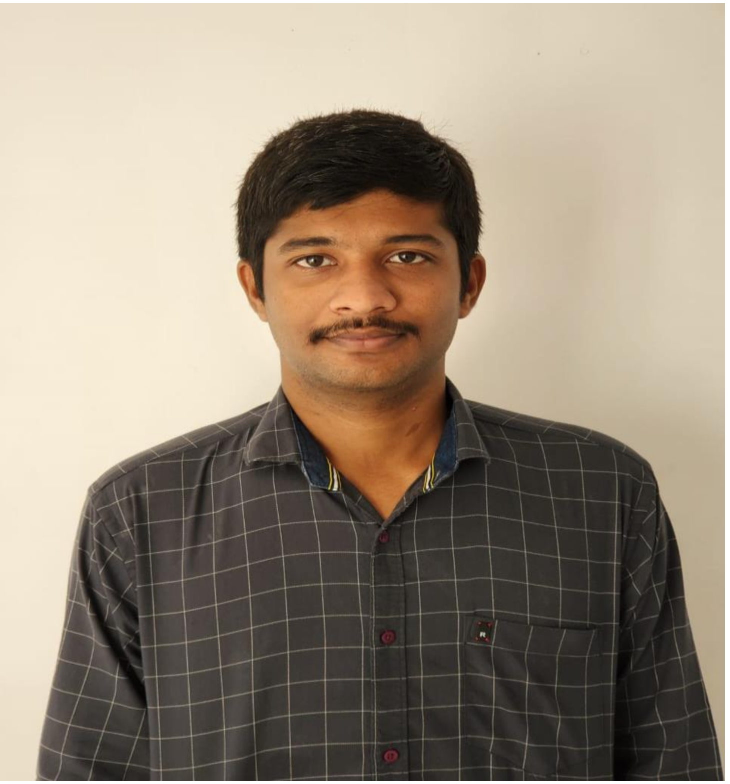


Fig 1: Encrypted Image before Code Execution

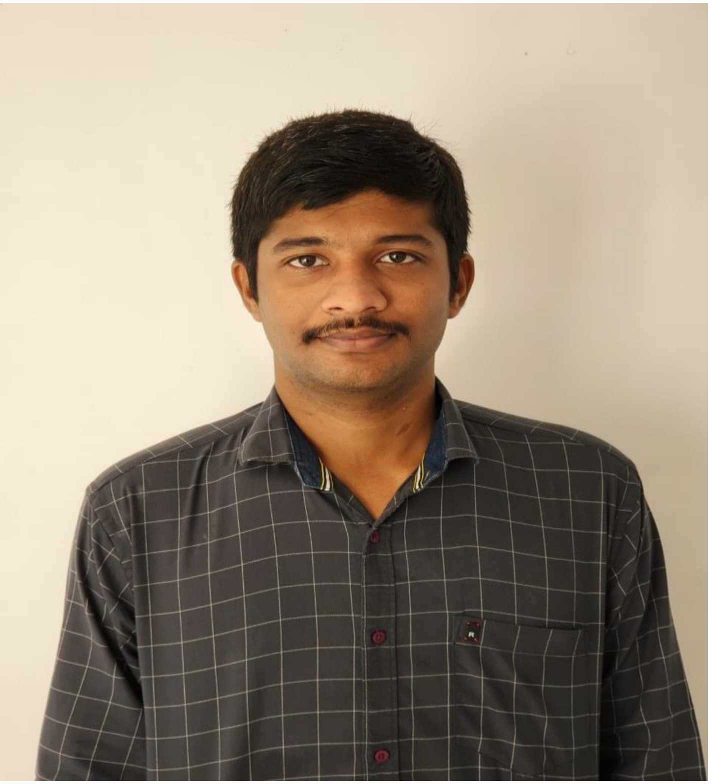
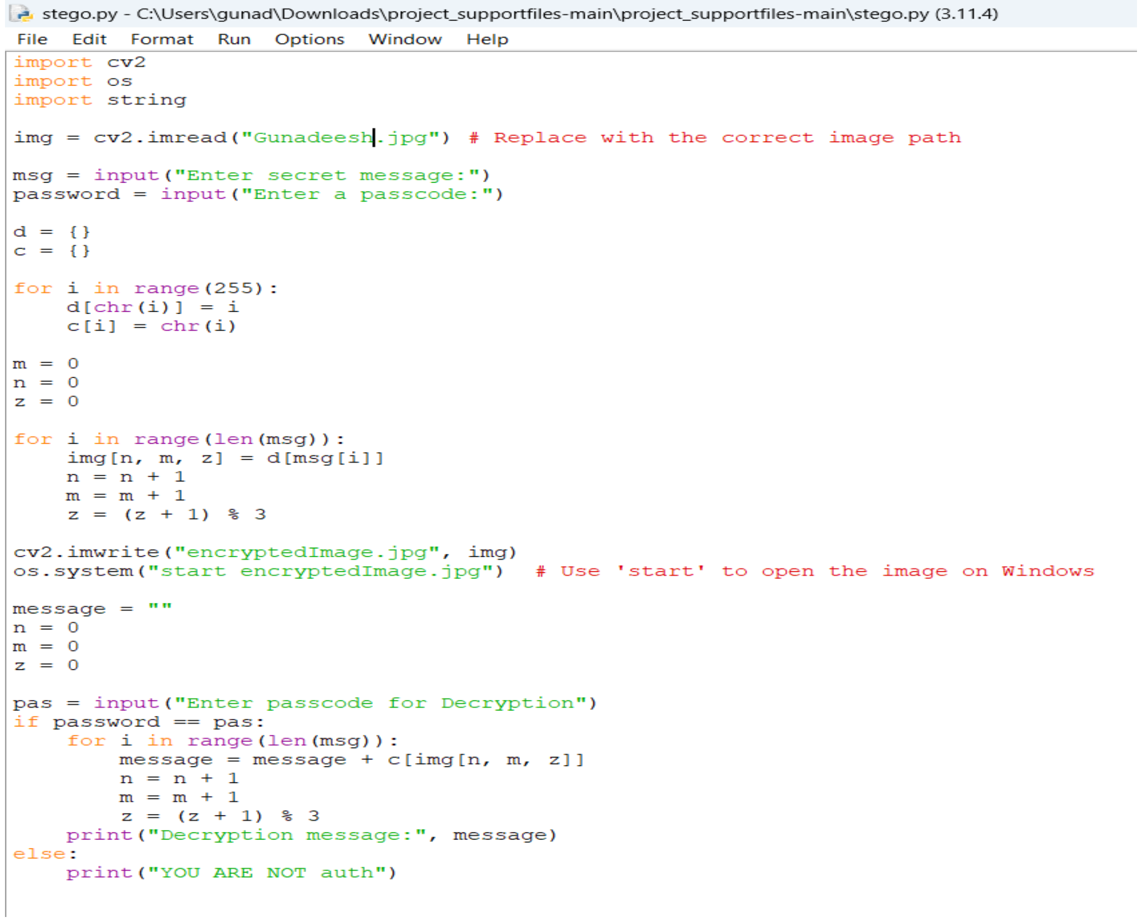
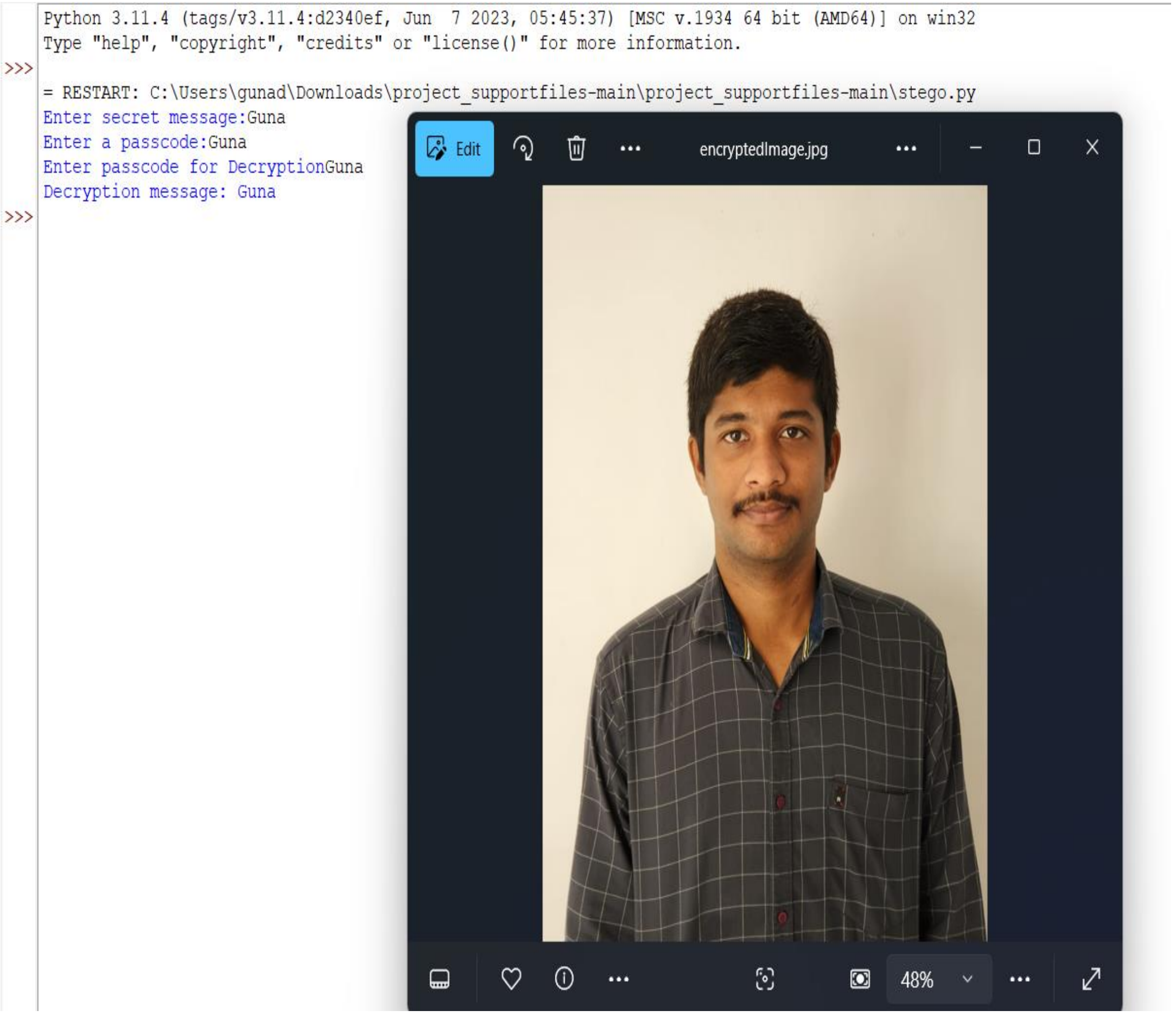


Fig 2: Encrypted Image after Code Execution

RESULTS


stego.py - C:\Users\gunad\Downloads\project_supportfiles-main\project_supportfiles-main\stego.py (3.11.4)
File Edit Format Run Options Window Help

```
import cv2
import os
import string

img = cv2.imread("Gunadeesh.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 Windows

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

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")
```



```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gunad\Downloads\project_supportfiles-main\project_supportfiles-main\stego.py
Enter secret message:Guna
Enter a passcode:Guna
Enter passcode for DecryptionGuna
Decryption message: Guna
>>>
```

INPUT : Encrypted Image before Code Execution

OUTPUT : Encrypted Image after Code Execution

RESULTS

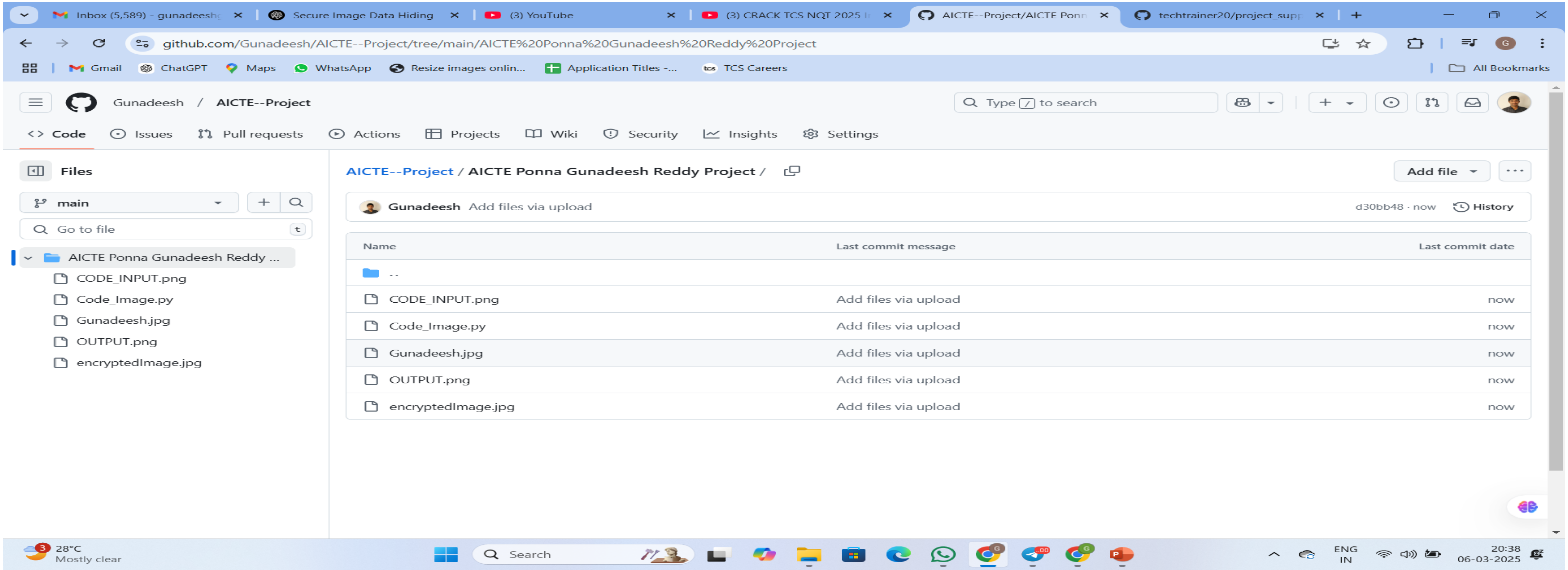
- The project successfully embeds encrypted text messages within images while maintaining high imperceptibility. Even after basic image modifications (such as compression and minor scaling), the hidden data remains recoverable using the correct decryption key.
- The system ensures that data retrieval is possible only by authorized users, making it highly secure against unauthorized access.

CONCLUSION

- This project effectively demonstrates how steganography, when combined with encryption, can enhance data security beyond conventional methods.
- The approach ensures that sensitive data can be hidden within images in a way that remains undetectable to third parties.
- The implementation provides a strong foundation for secure digital communication, proving useful in multiple domains.

GITHUB LINK

- https://github.com/Gunadeesh/AICTE--Project
- https://github.com/Gunadeesh/AICTE--Project/tree/main/AICTE%20Ponna%20Gunadeesh%20Reddy%20Project



github.com/Gunadeesh/AICTE--Project/tree/main/AICTE%20Ponna%20Gunadeesh%20Reddy%20Project

AICTE--Project / AICTE Ponna Gunadeesh Reddy Project /

Gunadeesh Add files via upload d30bb48 · now

Name	Last commit message	Last commit date
..		
CODE_INPUT.png	Add files via upload	now
Code_Image.py	Add files via upload	now
Gunadeesh.jpg	Add files via upload	now
OUTPUT.png	Add files via upload	now
encryptedImage.jpg	Add files via upload	now

FUTURE SCOPE(OPTIONAL)

- **AI-Based Steganalysis Resistance:** Implement machine learning models to improve resistance against steganalysis attacks.
- **Video & Audio Steganography:** Extend the concept to securely hide data within videos and audio files.
- **Cloud-Based Secure Sharing:** Integrate cloud storage options for secure and encrypted stego-image sharing.
- **Blockchain for Authentication:** Use blockchain technology to validate and track steganographic messages securely.
- **Enhanced Compression Handling:** Develop techniques to further reduce the impact of lossy compression on data integrity.

THANK YOU