NAME : Gautam Kumthekar

CLASS : TE A ROLL NO : 26

SUBJECT : NETWORK SECURITY

EXPERIMENT NO : IMPLEMENTATION OF STEGANOGRAPHY

CODE :

from PIL import Image

# Function to convert an integer to a binary string

def int\_to\_bin(n):

return bin(n)[2:].zfill(8)

# Function to convert a binary string to an integer

def bin\_to\_int(b):

return int(b, 2)

# Function to hide text message inside an image

def hide\_text\_in\_image(image\_path, text):

# Open the image

image = Image.open(image\_path)

# Convert the text to binary

binary\_text = ''.join([int\_to\_bin(ord(c)) for c in text])

# Check if the image is large enough to hold the text

max\_chars = (image.size[0] \* image.size[1] \* 3) // 8

if len(binary\_text) > max\_chars:

raise ValueError("Text too long to hide in the image.")

# Iterate over each pixel of the image

pixels = list(image.getdata())

encoded\_pixels = []

index = 0

for pixel in pixels:

if index < len(binary\_text):

# Modify the least significant bit of each color channel

r, g, b = pixel

r = (r & 0xFE) | int(binary\_text[index])

g = (g & 0xFE) | int(binary\_text[index + 1])

b = (b & 0xFE) | int(binary\_text[index + 2])

encoded\_pixels.append((r, g, b))

index += 3

else:

encoded\_pixels.append(pixel)

# Create a new image with the encoded pixels

encoded\_image = Image.new(image.mode, image.size)

encoded\_image.putdata(encoded\_pixels)

# Save the encoded image

encoded\_image.save("encoded\_image.png")

print("Text hidden in the image successfully.")

# Function to extract text message from an image

def extract\_text\_from\_image(image\_path):

# Open the image

encoded\_image = Image.open(image\_path)

# Extract the hidden text from the image

pixels = list(encoded\_image.getdata())

binary\_text = ''

for pixel in pixels:

# Extract the least significant bit of each color channel

r, g, b = pixel

binary\_text += str(r & 1)

binary\_text += str(g & 1)

binary\_text += str(b & 1)

# Convert the binary text to ASCII characters

text = ''

for i in range(0, len(binary\_text), 8):

byte = binary\_text[i:i + 8]

text += chr(bin\_to\_int(byte))

return text

# Example usage

image\_path = "image.png"

text\_to\_hide = "Hello, Steganography!"

# Hide the text in the image

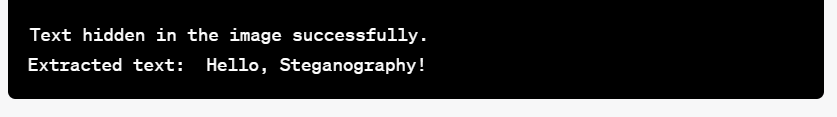
hide\_text\_in\_image(image\_path, text\_to\_hide)

# Extract the hidden text from the image

extracted\_text = extract\_text\_from\_image("encoded\_image.png")

print("Extracted text: ", extracted\_text)

**OUTPUT**

****