```python
import tkinter as tk
from tkinter import filedialog
from tkinter import messagebox
from PIL import Image, ImageTk
import numpy as np

def encrypt_image(image_path, key, output_path):
    img = Image.open(image_path).convert('RGB')
    pixels = np.array(img)

    height, width, _ = pixels.shape

    # Apply XOR operation with the key
    pixels ^= key

    # Swap pixel values in a simple pattern
    for i in range(height):
        for j in range(0, width, 2):
            if j + 1 < width:
                pixels[i, j], pixels[i, j + 1] = pixels[i, j + 1], pixels[i, j]

    encrypted_img = Image.fromarray(pixels)
    encrypted_img.save(output_path)

def decrypt_image(encrypted_image_path, key, output_path):
    encrypted_img = Image.open(encrypted_image_path).convert('RGB')
    pixels = np.array(encrypted_img)

    height, width, _ = pixels.shape
```

```
    # Swap pixel values back to their original positions
    for i in range(height):
        for j in range(0, width, 2):
            if j + 1 < width:
                pixels[i, j], pixels[i, j + 1] = pixels[i, j + 1], pixels[i, j]

    # Apply XOR operation with the key to decrypt
    pixels ^= key

    decrypted_img = Image.fromarray(pixels)
    decrypted_img.save(output_path)

def select_image():
    file_path = filedialog.askopenfilename(filetypes=[("Image files", "*.jpg *.png *.jpeg")])
    if file_path:
        image_path.set(file_path)
        img = Image.open(file_path)
        img.thumbnail((200, 200))
        img = ImageTk.PhotoImage(img)
        panel.config(image=img)
        panel.image = img

def encrypt():
    if not image_path.get():
        messagebox.showwarning("Warning", "Please select an image")
        return

    key = key_entry.get()
    if not key.isdigit():
        messagebox.showwarning("Warning", "Key must be a number")
        return
```

```
    key = key_entry.get()
    if not key.isdigit():
        messagebox.showwarning("Warning", "Key must be a number")
        return

    key = int(key) % 256
    output_path = filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG files", "*.png")])
    if output_path:
        encrypt_image(image_path.get(), key, output_path)
        messagebox.showinfo("Success", "Image encrypted successfully")

def decrypt():
    if not image_path.get():
        messagebox.showwarning("Warning", "Please select an image")
        return

    key = key_entry.get()
    if not key.isdigit():
        messagebox.showwarning("Warning", "Key must be a number")
        return

    key = int(key) % 256
    output_path = filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG files", "*.png")])
    if output_path:
        decrypt_image(image_path.get(), key, output_path)
        messagebox.showinfo("Success", "Image decrypted successfully")

# Setting up the GUI
root = tk.Tk()
root.title("Image Encryption Tool")

image_path = tk.StringVar()

frame = tk.Frame(root)
frame.pack(pady=10)
```

```
# Setting up the GUI
root = tk.Tk()
root.title("Image Encryption Tool")

image_path = tk.StringVar()

frame = tk.Frame(root)
frame.pack(pady=10)

panel = tk.Label(frame)
panel.pack()

select_button = tk.Button(frame, text="Select Image", command=select_image)
select_button.pack(pady=5)

key_label = tk.Label(frame, text="Key (0-255):")
key_label.pack(pady=5)

key_entry = tk.Entry(frame)
key_entry.pack(pady=5)

encrypt_button = tk.Button(frame, text="Encrypt Image", command=encrypt)
encrypt_button.pack(pady=5)

decrypt_button = tk.Button(frame, text="Decrypt Image", command=decrypt)
decrypt_button.pack(pady=5)

root.mainloop()
```

Image Encryption Tool

Select Image

Key (0-255):

150

Encrypt Image

Decrypt Image

92°F
Haze

Search

ENG
IN

13:57
21-07-2024

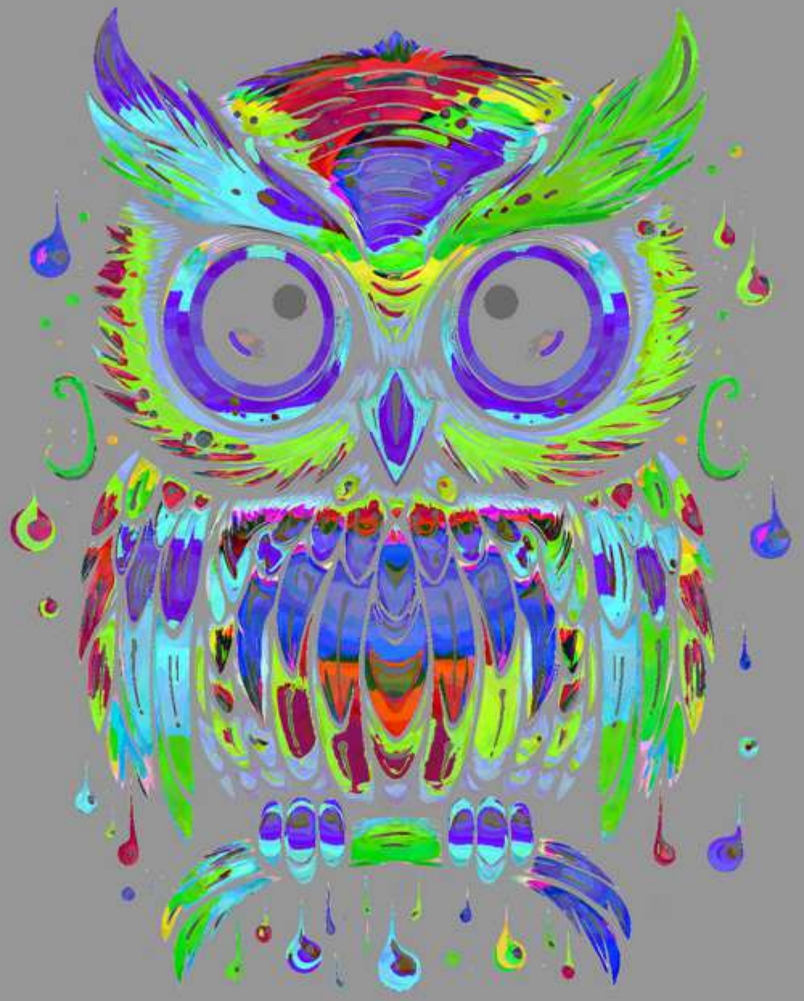

Image Encryption Tool

Select Image

Key (0-255):

150

Encrypt Image

Decrypt Image