|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| !pip install pillow   |  |  | | --- | --- | | Requirement already satisfied: pillow i | n | | /usr/local/lib/python3.11/dist-packages (11.1.0) | |   from google.colab import files  *# Upload the file* uploaded = files.upload()   |  |  | | --- | --- | | <IPython.core.display.HTML object> | | | Saving input.jpg to input (2).jpg |   input\_image = "input.jpg" *# Replace with your uploaded image file name*  encrypted\_image = "encrypted.jpg" *# Path to save the encrypted image* decrypted\_image = "decrypted.jpg" *# Path to save the decrypted image* from PIL import Image  *# Encryption Function* def encrypt\_image(input\_path, output\_path): img = Image.open(input\_path) pixels = img.load() width, height = img.size  for i in range(width): for j in range(height): r, g, b = pixels[i, j]  *# Swap Red and Blue channels*  encrypted\_pixel = (b, g, r) pixels[i, j] = encrypted\_pixel  img.save(output\_path)  print(f"Image encrypted successfully! Saved as: {output\_path}")  *# Decryption Function with Color Change* def decrypt\_image\_with\_color\_change(input\_path, output\_path): img = Image.open(input\_path) pixels = img.load() width, height = img.size  for i in range(width): for j in range(height): r, g, b = pixels[i, j]  *# Swap back Blue and Red*  original\_pixel = (b, g, r) |

|  |  |  |
| --- | --- | --- |
| *# Apply a color transformation (e.g., increase red by 50, cap at 255)*  new\_r = min(original\_pixel[0] + 50, 255) *# Increase red*  new\_g = max(original\_pixel[1] - 30, 0) *# Decrease green*  new\_b = original\_pixel[2] *# Keep blue unchanged*  pixels[i, j] = (new\_r, new\_g, new\_b)  img.save(output\_path) print(f"Image decrypted with color change! Saved as:  {output\_path}")  encrypt\_image(input\_image, encrypted\_image)   |  | | --- | | Image encrypted successfully! Saved as: encrypted.jpg |   decrypt\_image\_with\_color\_change(encrypted\_image, decrypted\_image)   |  | | --- | | Image decrypted with color change! Saved as: decrypted.jpg |   from IPython.display import display  *# Display Original Image* print("Original Image:") display(Image.open(input\_image))  *# Display Encrypted Image* print("Encrypted Image:")  display(Image.open(encrypted\_image))  *# Display Decrypted Image with Color Change* print("Decrypted Image with Color Change:") display(Image.open(decrypted\_image)) Original Image: |



Encrypted Image:



Decrypted Image with Color Change:

