**from PIL import Image**

**def int\_to\_bin(rgb):**

**"""Convert an integer tuple to a binary (string) tuple."""**

**r, g, b = rgb**

**return ('{0:08b}'.format(r),**

**'{0:08b}'.format(g),**

**'{0:08b}'.format(b))**

**def encode(img, secret\_file):**

**"""Encode the secret file within the image."""**

**image = Image.open(img)**

**secret = open(secret\_file, 'rb').read()**

**if len(secret) \* 8 > image.width \* image.height \* 3:**

**raise ValueError("Input file too large to encode in image.")**

**index = 0**

**secret += b"\0" \* (len(image.tobytes()) - len(secret)) # Pad secret to match image size**

**for x in range(image.width):**

**for y in range(image.height):**

**if index < len(secret):**

**pixel = list(image.getpixel((x, y)))**

**bin\_r, bin\_g, bin\_b = int\_to\_bin(pixel)**

**bin\_r = bin\_r[:-1] + '{0:01b}'.format(secret[index] >> 7 & 1)**

**bin\_g = bin\_g[:-1] + '{0:01b}'.format(secret[index] >> 6 & 1)**

**bin\_b = bin\_b[:-1] + '{0:01b}'.format(secret[index] >> 5 & 1)**

**pixel = tuple([int(bin\_r, 2), int(bin\_g, 2), int(bin\_b, 2)])**

**image.putpixel((x, y), pixel)**

**index += 1**

**image.save('encoded\_image.png')**

**print("Encoding successful.")**

**def decode(img):**

**"""Decode the secret file from the image."""**

**image = Image.open(img)**

**binary\_data = ""**

**for x in range(image.width):**

**for y in range(image.height):**

**r, g, b = int\_to\_bin(image.getpixel((x, y)))**

**binary\_data += r[-1] + g[-1] + b[-1]**

**secret = bytes([int(binary\_data[i:i + 8], 2) for i in range(0, len(binary\_data), 8)])**

**with open('decoded\_secret.txt', 'wb') as f:**

**f.write(secret)**

**print("Decoding successful.")**

**# Example usage**

**encode('original\_image.png', 'secret\_file.txt')**

**decode('encoded\_image.png')**