

A3

ESTEGANOGRAFIA

Aitor Sánchez Salgado

DAM2 2024/25

Contingut:

Amagar missatge en una Imatge.....	3
Revelar missatge d'una Imatge	5
1a funció:.....	5
2a funció.....	6
3a funció.....	6

Amagar missatge en una Imatge

```
def hide_message():
    selectedPath = load_image.file_path
    if not selectedPath:
        print("No file selected")
        return
    try:
        img = Image.open(selectedPath)
        img = img.convert("RGB")
    except Exception as e:
        print(f"Error opening image: {e}")
        return

    message = text_area.get("1.0", tk.END).strip()
    key = keyTextArea.get("1.0", tk.END).strip()
    if not key:
        messagebox.showinfo("No key", "A key is needed")
        return
    message += f"-/-/{key}-/-/-by {current_user}" + chr(0)
    binary_message = ''.join(format(ord(char), '08b') for char in message)

    required_bits = len(binary_message)
    width, height = img.size
    available_pixels = width * height

    if required_bits > available_pixels * 3:
        messagebox.showerror("Error", "The message is too large to fit in this image.")
        return

    message_index = 0
    pixel_count = 0
    for x in range(width):
        for y in range(height):
            if message_index < len(binary_message):
                r, g, b = img.getpixel((x, y))

                for color in range(3):
                    if message_index < len(binary_message):
                        bit_value = int(binary_message[message_index])
                        if color == 0:
                            r = (r // 2 * 2) + bit_value
                        elif color == 1:
                            g = (g // 2 * 2) + bit_value
                        else:
                            b = (b // 2 * 2) + bit_value
                        message_index += 1
                    else:
                        break
                pixel_count += 1
            else:
                break
        else:
            break
```

```
        elif color == 2:
            b = (b // 2 * 2) + bit_value
            message_index += 1

    img.putpixel((x, y), (r, g, b))
    pixel_count += 1

    if message_index >= len(binary_message):
        break

modified_image_path = 'hidden_message_image.png'
img.save(modified_image_path)
```

Revelar missatge d'una Imatge

Dividit en 3 fun:

1a funció:

S'utilitza per a demanar la imatge que vol obrir amb el missatge i demanar la contrasenya del missatge.

Processa el missatge i elimina la key del missatge. Si el missatge existeix, el mostra amb una message box.

```
def getMessageFromImage():
    file_path = filedialog.askopenfilename(title="Select an Image")

    if file_path:
        try:
            hidden_message = reveal_message(file_path)
            usrKey = simpledialog.askstring("Key", "Enter the key of the
message")

            if hidden_message:
                spt = hidden_message.split("-/-/-")
                if(spt[len(spt)-2] != usrKey):
                    messagebox.showerror("Incorrect key", "The key given
is incorrect")

                    return
                if(len(spt) > 3):
                    for i in range(0, len(spt)-2):
                        hidden_message += spt[i] + "-/-/-"
                    hidden_message += "\n"+spt[len(spt)-1]
                else:
                    hidden_message = spt[0] + "\n" + spt[2]

                messagebox.showinfo("Hidden Message", hidden_message)
            else:
                messagebox.showwarning("No Message Found", "No hidden
message was found in the image.")
        except Exception as e:
            print(e)
```

2a funció

Revela el missatge fent la inversa. Agafa els bits menys significants i els afegeix en una cadena de bits. Aquesta, la comprovo constantment per a veure si els últims 8 bits son la marca final del missatge. Si no troba marca final, retorna els últims bits de la imatge sencera com a missatge.

```
def reveal_message(file_path):
    img = Image.open(file_path)
    width,height = img.size
    binary_message = ''
    for x in range(width):
        for y in range(height):
            r,g,b = img.getpixel((x, y))
            for i in range(0,3):
                if i == 0:
                    binary_message += str(r & 1)
                elif i == 1:
                    binary_message += str(g & 1)
                elif i == 2:
                    binary_message += str(b & 1)

            if len(binary_message) >= 8 and len(binary_message) % 8 == 0:
                last_byte = binary_message[-8:]
                if chr(int(last_byte, 2)) == chr(0):
                    return convertToString(binary_message[:-8])
```

3a funció

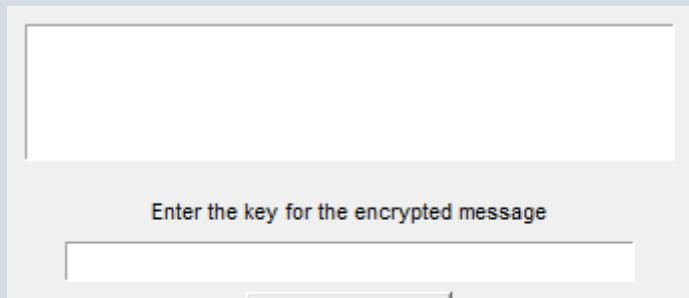
Amb la seqüència de bits que m'envien, la divideixo de 8 en 8 per a convertir-ho en chars a través de l'ascii. Si el byte no té 8 bits o no es pot transformar en char, el salto i passo al següent.

```
def convertToString(binaryString):
    chars = []
    for i in range(0, len(binaryString), 8):
        byte = binaryString[i:i + 8]
        if len(byte) < 8:
            continue
        try:
            char = chr(int(byte, 2))
            if char == chr(0):
                break
            chars.append(char)
        except ValueError:
            continue

    return ''.join(chars)
```

Key personalitzada

He afegit un requadre sota el del missatge on et demana una key per al missatge a l'hora de encriptar el missatge. Si aquesta és buida, salta una messagebox d'error demanant la key.



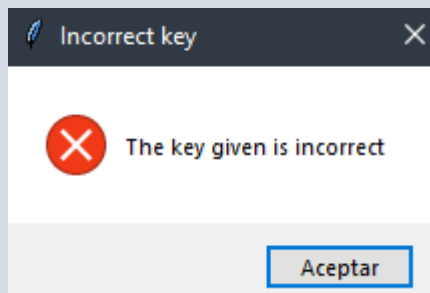
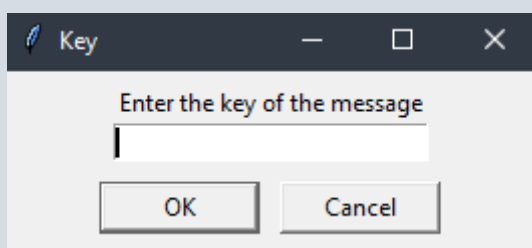
Implementació de la TextArea i la Label de sobre

```
# Text area i botó "Amagar missatge" (ocult)
text_area = tk.Text(app, height=4, width=40)
keyLbl = tk.Label(app, text="Enter the key for the encrypted message",
font=("Arial", 8))
keyTextArea = tk.Text(app, height=1, width=35)
hide_message_button = tk.Button(app, text="Amagar missatge",
command=hide_message)
```

Error si la key és buida

```
key = keyTextArea.get("1.0", tk.END).strip()
if not key:
    messagebox.showinfo("No key", "A key is needed")
    return
```

A l'hora de revelar el missatge, demano la key amb un simpledialog el qual invoco quan s'ha seleccionat la imatge que es vol desencriptar. Desencripto el missatge i, si la key no coincideix, no el mostro i ensenyo una messagebox d'error dient que la key és incorrecta.



SimpleDialogue per la key

```
usrKey = simpledialog.askstring("Key", "Enter the key of the message")
```

Error amb la key incorrecta

```
spt = hidden_message.split("-/-/-")
if(spt[len(spt)-2] != usrKey):
    messagebox.showerror("Incorrect key", "The key given is incorrect")
    return
```