**A3**

Aitor Sánchez Salgado

DAM2 2024/25

**ESTEGANOGRAFIA**

Contingut:

[Amagar missatge en una Imatge 3](#_Toc179388275)

[Revelar missatge d'una Imatge 5](#_Toc179388276)

[1a funció: 5](#_Toc179388277)

[2a funció 6](#_Toc179388278)

[3a funció 6](#_Toc179388279)

# 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

                        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)

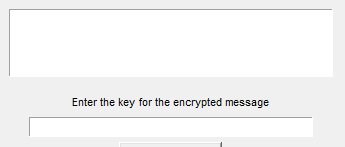
        except ValueError:

            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

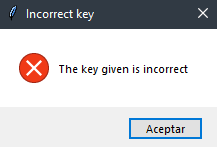
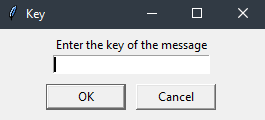
    key = keyTextArea.get("1.0", tk.END).strip()

    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