

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Ciencias

Proyecto de Modelado y Programación

Asignatura: Modelado y Programación

Alumnos:

- Urzua Contreras Cristian Josué - Líder
- Rosas Marín Jesús Martín - BackEnd
- Reyes Arteaga Ángel David - FrontEnd

Profesores:

- José de Jesús Galaviz Casas
- María Ximena Lezama Hernández
- Karla Adriana Esquivel Guzmán

Proyecto de Modelado y Programación

1 Descripción del Programa

Este programa permite ocultar y revelar mensajes de texto en archivos de imagen mediante el método de esteganografía basado en el bit menos significativo (LSB). Los usuarios pueden usar este programa para ocultar mensajes en una imagen y luego extraerlos cuando sea necesario.

1.1 Requisitos

- Python 3.x
- Biblioteca Pillow para manipulación de imágenes

Instala Pillow ejecutando:

```
pip install Pillow
```

2 Estructura del Código

1. **text_to_binary(text)**: Convierte un mensaje de texto en una cadena binaria.
2. **hide_message(image_path, message, salida)**: Oculta un mensaje de texto en una imagen.
3. **reveal_message(image_path)**: Extrae el mensaje oculto de una imagen.
4. **Ejecución desde línea de comandos**: La función `--main--` interpreta los argumentos proporcionados.

3 Funcionalidades

3.1 Función `text_to_binary(text)`

Convierte el texto en una representación binaria, permitiendo que cada carácter del mensaje sea convertido a su correspondiente código ASCII en binario.

Ejemplo de uso:

```
text_to_binary("Hola")
```

3.2 Función `hide_message(image_path, message, salida)`

Oculta el mensaje en una imagen:

- Convierte el mensaje en binario y añade una longitud de encabezado (16 bits) y un carácter de terminación ('00000000').
- Verifica si la imagen tiene suficiente capacidad para almacenar el mensaje.
- Recorre cada píxel de la imagen y ajusta el bit menos significativo de cada canal RGB para almacenar el mensaje.
- Guarda la imagen con el mensaje oculto.

Parámetros:

- `image_path`: Ruta de la imagen en la que se ocultará el mensaje.
- `message`: Texto a ocultar.
- `salida`: Nombre del archivo de salida para la imagen con el mensaje oculto.

Ejemplo de uso:

```
hide_message("imagen.png", "Mensaje secreto", "imagen_oculta")
```

3.3 Función `reveal_message(image_path)`

Revela el mensaje oculto en una imagen:

- Extrae los bits de los canales RGB de cada píxel.
- Decodifica el mensaje binario, comenzando con el encabezado que indica la longitud del mensaje.
- Si falta el carácter de terminación, se imprime una advertencia.

Parámetro:

- `image_path`: Ruta de la imagen que contiene el mensaje oculto.

Ejemplo de uso:

```
reveal_message("imagen_oculta.png")
```

4 Ejecución desde Línea de Comandos

El programa acepta dos modos de operación:

- **Ocultar mensaje (-h)**: Oculta un mensaje en la imagen.
- **Revelar mensaje (-u)**: Extrae el mensaje oculto de una imagen.

4.1 Opciones de Ejecución

4.1.1 Opción -h (Ocultar mensaje)

```
python nombre_programa.py -h archivo_texto imagen_entrada nombre_imagen_salida
```

- `archivo_texto`: Nombre del archivo que contiene el texto a ocultar.
- `imagen_entrada`: Imagen en la que se va a ocultar el mensaje.
- `nombre_imagen_salida`: Nombre de la imagen resultante con el mensaje oculto.

Ejemplo:

```
python esteganografia.py -h mensaje.txt imagen.png imagen_oculta
```

4.1.2 Opción -u (Revelar mensaje)

```
python nombre_programa.py -u imagen_entrada archivo_texto_salida
```

- `imagen_entrada`: Imagen que contiene el mensaje oculto.
- `archivo_texto_salida`: Archivo donde se guardará el mensaje revelado.

Ejemplo:

```
python esteganografia.py -u imagen_oculta.png mensaje_revelado.txt
```

5 Ejemplos de Uso

- Ocultar un mensaje:

```
python esteganografia.py -h mensaje.txt imagen.png imagen_oculta
```

Esto ocultará el contenido de `mensaje.txt` en `imagen.png` y generará una nueva imagen llamada `imagen_oculta.png` con el mensaje oculto.

- Revelar un mensaje:

```
python esteganografia.py -u imagen_oculta.png mensaje_revelado.txt
```

Esto extraerá el mensaje oculto en `imagen_oculta.png` y lo guardará en `mensaje_revelado.txt`.

6 Precauciones y Advertencias

- El archivo de imagen debe ser lo suficientemente grande para almacenar el mensaje deseado.
- Evita guardar imágenes con mensajes ocultos en formatos de compresión con pérdida (por ejemplo, JPEG), ya que esto puede alterar el mensaje oculto.