

Steganography Project

The knowledge that a secret exists is half of the secret -- Joshua Meyrowitz

What is Steganography?

From the Greek words *στεγανός* and *γράφω* meaning “covered writing.”

- Art and Science of hiding information in ways that prevent detection.
- Can be used on audio, text, packet headers, or images
- Similar to Cryptography –Cryptography: scrambles message –Steganography: hides message

History

In his history of the Persian Wars, Herodotus tells of a messenger who shaved his head and allowed a secret message to be tattooed on his scalp. He waited until his hair grew back. Then he journeyed to where the recipient awaited him and shaved his head again. The message was revealed. It was history’s first use of steganography.

Hiding a picture within a picture

Least Significant Bit (LSB)

- One of most common techniques
- Alters LSB of each pixel (1 bit out of 24 --or 1 out of 8 for grayscale)
- Uses the concept of *parity*, ie, even numbers in binary end in 0, odd ones end in 1
- Easiest to implement: hiding bitmaps in a color picture
- Hiding ASCII code, one letter at a time

Example:

To hide letter C (**100011**) in a grayscale file:

Original:

01001101 01001110 01001110 01001111 01010000 01010000 01001111

1 0 0 0 0 1 1

Encoded:

01001101 01001110 01001110 01001110 01010000 01010001 01001111

Encoding a message

The function `encode()` creates an image with message written in big black letters across it; it then invokes `encodePicture()` to hide the image with the message in the given picture

```

def encode(picture, message):
    import java.awt.Font as Font
    myFont = makeStyle("Arial", Font.BOLD, 24)
    msgPicture = makeEmptyPicture(getWidth(picture),
                                  getHeight(picture),
                                  white)
    addTextWithStyle(msgPicture,10,28,message, myFont, black)
    encodePicture(picture, msgPicture)

def encodePicture(picture, msgPicture):
    for pxl in getPixels(picture):
        if (getRed(pxl) % 2) == 1:
            setRed(pxl, getRed(pxl) - 1)
    for x in range(0, getWidth(picture)):
        for y in range(0, getHeight(picture)):
            msgPxl = getPixel(msgPicture, x, y)
            origPxl = getPixel(picture, x, y)
            if distance(getColor(msgPxl), black) < 100.0:
                setRed(origPxl, getRed(origPxl)+1)

```

Assignment

Create a menu-driven picture manipulation set of functions that allow the user to make various changes to images. Here is the *algorithm*:

- welcome the user to your program
 - get the user to select a picture to work on
 - show(picture)
 - ask the user to select one of the following options:
 1. Make picture grayscale
 2. Decrease red by 10%
 3. Decrease green by 10%
 4. Decrease blue by 10%
 5. "Posterize"
 6. Encode a message
 7. Decode message
 8. <a feature of your choice>
 - repaint(picture)
 - ask user whether to save the picture work (enter y/n), and if the response is "y", save it
 - show goodbye/thank you message
-

Links

- [More examples](#) - you can copy and paste these to save time
- Test your function by decoding the hidden messages [in these images](#)
- Small images you can use to test your program: [eye.jpg](#), [luca.jpg](#)
- [Lecture slides](#)