**Steganography by using Least Significant Bit**

This uses a symmetric key algorithm to hide the data in an image and extract the hidden data from the image.

This program contains 2 parts -

1. Encoding - This part hides the user input message in an image. It uses the user-defined key for deciding the location of the pixels.
2. Decoding - This part extracts the hidden message from the image. generated from part-1, however, we need the same key to extract the message.

**Pre-requisite -**

1. Python-3
2. OpenCV

**Inputs -**

1. **Name of the image -** Along with extension - which is to be used to hide the message (for example - Cover\_1.png) Any image can be used as a carrier but the file should be kept in the same folder as the .pynb file)
2. **Key -** user-defined key (alpha-numeric)
3. **Message -** The alpha-numeric message which is to be hidden in the image

**Encoding -**

*encodingMessage(imageName, key, message)*

The encoding function takes 3 inputs - image, key, message. The encoded image is stored in the drive in the same folder as the .pynb file with the new filename - encodedImage.png.

**Decoding -**

*extractData(key)*

The decoding function takes 1 input - key. This function returns the extracted message.

**Run -**

Run the cell results and provide the input to run the program.