# Text Steganography in a BMP file

## Alan Le

My implementation of LSB image steganography is written in C++ and uses the EasyBMP library to handle the BMP image format. This program stores an array of type char in an output 24-bit image. Every char is embedded amongst the RGB values of two adjacent pixels. Output carrier files look identical to the input files to the human eye. The maximum possible amount of characters stored is the number of pixels in the carrier BMP file divided by two.

Some limitations of this project are that it can only properly decode 24-bit bitmaps that were encoded with this exact method, and can only handle alphanumeric characters. Further work on this would include creating a user-friendly GUI, integrating a steganalysis tool for decoding of other steganography files, and allowing for other embedding techniques (i.e. not limited to sequential bit encoding, support for other bitmap formats like 32-bit, encryption, bit plane steganography).

The result of this program (LSB Steganography) is not particularly secure enough against statistical steganalysis.

Operation:

1. Use the makefile to compile the code (I compiled the code on thunder.cise.ufl.edu)
2. Run the executable. The first input argument should be “e” or “d” for encoding and decoding specifically.

If encoding, the second input argument is the text string, the third input argument should be the carrier bmp file, and the fourth input argument should be the output bmp. Example: ./program e “this is the string” dogs.bmp output.bmp

If decoding, the second input argument is the bmp file to decode. Example: ./program d output.bmp

Successful Test Cases:

Encrypting with the maximum amount of characters (TotalPixels /2). The program successfully truncates extra characters.

Can take in various BMP types (256 color, 24-bit, 32-bit) and successfully operate.

Incorrect input arguments give error messages, such as pointing to improper BMP files (handled by EasyBMP) or not specifying “encode” or “decode”.

Files:

stego.cpp

Makefile

(EasyBMP library version 1.06 <http://easybmp.sourceforge.net/download.html>)

EasyBMP.cpp

EasyBMP.h

EasyBMP\_BMP.h

EasyBMP\_DataStructures.h

EasyBMP\_VariousBMPutilities.h

## References: (from presentation)

Fridrich, J., Goljan, M., & Du, R. (n.d.). **Detecting LSB steganography in color, and gray-scale images.** *IEEE Multimedia,* 22-28.

Ker, A. (2004). **Quantitative evaluation of pairs and RS steganalysis.** *Security, Steganography, and Watermarking of Multimedia Contents VI*.

Rodriguez, B., Peterson, G., & Bauer, K. (n.d.). **Fusion of Steganalysis Systems Using Bayesian Model Averaging**. *IFIP — The International Federation for Information Processing Advances in Digital Forensics IV,* 345-355.

El\_Rahman, S. (2015). **A Comprehensive Image Steganography Tool using LSB Scheme**. *International Journal of Image, Graphics and Signal Processing IJIGSP,* 10-18.

Newman, R., Moskowitz, I., Chang, L., & Brahmadesam, M. (2002). **A Steganographic Embedding Undetectable by JPEG Compatibility Steganalysis**. *Information Hiding Lecture Notes in Computer Science,* 258-277.

Boehm, B. (2014). **StegExpose - A Tool for Detecting LSB Steganography.** http://arxiv.org/pdf/1410.6656v1.pdf

Westfeld, A., & Pfitzmann, A. (n.d.). **Attacks on Steganographic Systems**. *Information Hiding Lecture Notes in Computer Science,* 61-76.

Macklin, P. **EasyBMP.** http://easybmp.sourceforge.net/

Fridrich, J., Goljan, M., & Du, R. (n.d.). **Reliable detection of LSB steganography in color and grayscale images**. *Proceedings of the 2001 Workshop on Multimedia and Security New Challenges - MM&Sec '01*.