TJHSST Computer Systems Lab Senior Research Project
Implementation of Least Significant Bit Steganography and its Steganalysis
2009-2010

Deniz Oran

June 15, 2010

Abstract

Although steganography is an ancient form of unobtrusive and covert communications, its implementation has become significantly more efficient due to the advent of image storage technology and the associated amounts of space available to encode a text message. An implementation of Least Significant Bit (LSB) steganographical techniques covertly encodes any text communication into a .PNG image file without causing the image to appear different than the uncompromised image to the naked human eye. The JAVA programming language is utilized to exploit the inherent "noise" or data that does not significantly contribute to the image. The program also features the LSBs steganalysis or the decoding of an image encoded using the program. The text decoded was found to be identical to the text originally entered by the user encoding the communication. The features of the program have then been integrated into a Graphical User Interface (GUI) for ease of use and aesthetic appeal.

Keywords: Steganography, Steganalysis, Least Significant Bit encoding, image carrier

1 Introduction - Elaboration on the problem statement, purpose, and project scope

1.1 Scope of Study

Steganography has evolved historically with the digital age. More and more data is being sent to other computers through the internet and an inexpensive method of covertly sending that data was discovered. Encryption is typically expensive in terms of purchasing software and the computing power needed to encrypt and decrypt at the other end. Steganography

is rapidly evolving as an alternative to encrpytion and is considered a possible method of espionage and cyber-terrorism. Research in the field has rapidly accelerated to deter the misuse of steganographical encoding techniques. This program will be an introduction to the field of steganography because it uses one of the fundamental encoding methods.

1.2 Type of research

This project would be user-inspired research because the ultimate goal is to effectively use the software in real-world communications. The program is not going to utilize something such as a command prompt because a GUI would better facilitate usability. Integrating the encoding and extraction code with the GUI is necessary for this program to be successful. There are numerous variations of steganography available, but the most practical application is within image steganography.

2 Background and review of current literature and research

The art and science of steganography has existed since the time of the Ancient Greeks and continues to be used for secure and covert communication. The initial method of unobtrusively concealing a sent message was to shave the hair off of a servant, inscribe the desired message, wait for the hair to grow back, and then send the servant to the recipient. A method that involved writing a on a wax tablet, covering it with fresh wax, and sending it by public means to therecipient was an even more progressive and economical method of secret communication. Steganography is therefore the opposite of encryption, in which a message is made unintelligible, but openly transmitted through public means. The concealment of the fact that a message is even being transferred is the true essence of steganography. The classic struggle of covertly communicating is best illustrated through the prisoners' dilemma, in which two prisoners must exchange plans to escape without provoking the suspicion of the warden. Encryption would be an ineffective method in this situation because as soon as the prison warden doesn't comprehend the communication he immediately shuts down all forms of communication between the prisoners, thus foiling their attempt at escaping.

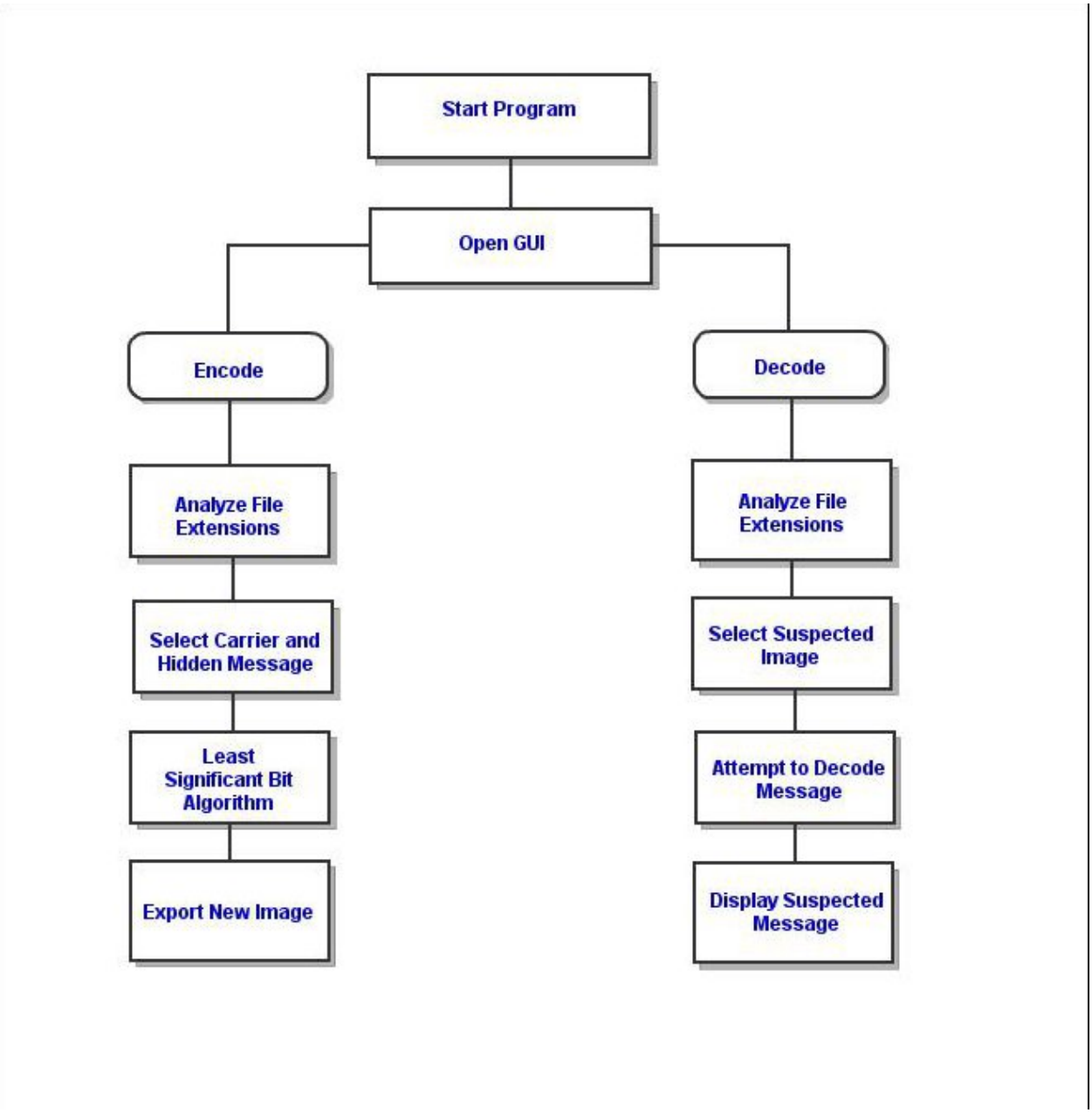
The other aspect of the program is the steganalysis of the encoded message. There is no current universal method of extracting a hidden message; the type of encoding used must be known. Even if it is known, LSB encoding varies with the coder and the original image will not be available to statistically compare with the suspected compromised image. This method was found to be 80 percent accurate in determining if a message is being encoded (not decoding the message). As shown, this method is increasingly complex and doesnt even lead to deciphering the message. Another method to extract the message from the compromised image must be found.

3 Procedures and Methodology

The JAVA programming language will be used because of its secure and stable structure along with its popularity. The most effective and most common way of covertly encoding a message or image is by hiding it in unused portions of commonly used lossless image formats such as .GIF or .PNG. To do this, a technique called LSB encoding is employed to edit numerous picayune parts of the image and placing parts of binary code that can be compiled by the reader to form a text message. The program therefore reads the entire binary composition of the image into a cubic data structure or three-dimensional array storing the pixel row, column, and color value by using the Writable Raster and DataBufferByte classes built into JAVA. The least significant bits are then found and replaced by the desired 1 or 0 by shifting the values of the byte and inserting the digit and deleting the previous one. The length of the message is then encoded before the message within the first few bytes of the image. The new image binary is then compiled together into a new matrix that can be saved in the host directory.

The second portion of the project is detecting the steganography and extracting the hidden message. Clearly the first test would simply be visually ensuring that the original image and the carrier appear identical. The second, and more sophisticated method, is through attempting to reverse engineer the encoding method. According to the previous research, there is no way to extract a message from an image without having an idea about how its encoded. This method is usually guessed by commercial software because of the prevalence of LSB encoding. To deter this, though, techniques such as inserting the information not at the beginning can be implemented. A commercial steganalysis program will be incapable of detecting this particular version of the encoding because the image wasnt generated by that program. Regardless, if the image is suspected, the uncompromised carrier will be required to compare the images bit by bit and having both the original image and the altered image is highly unlikely.

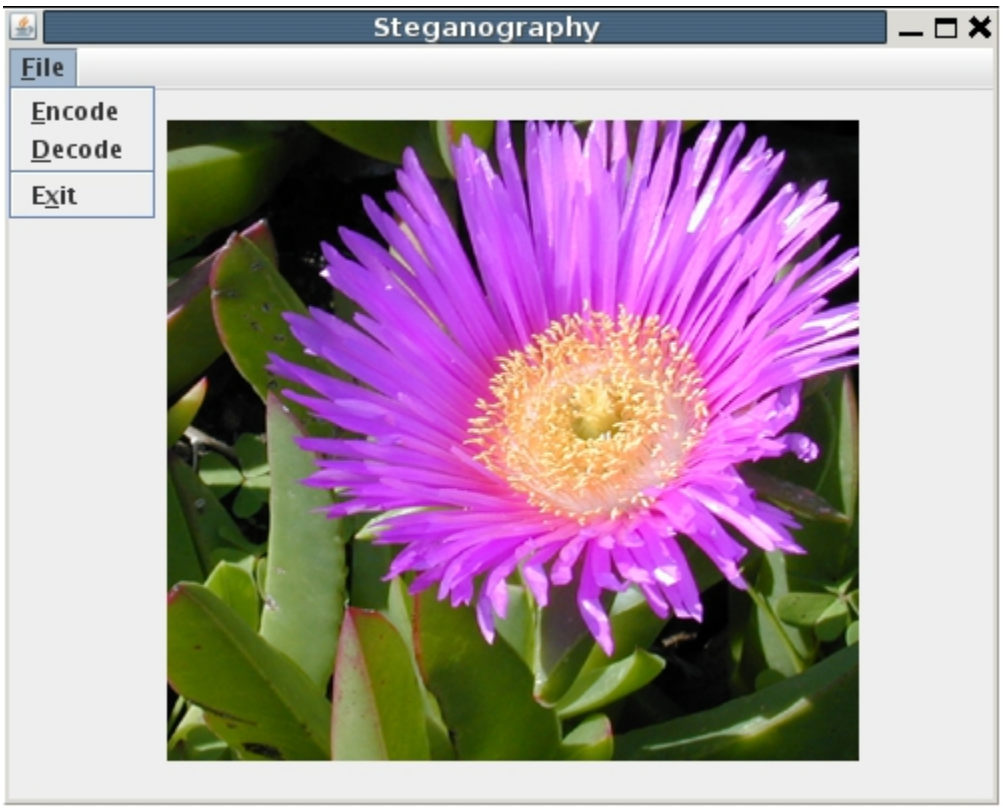
The decoding method used essentially reversed what was done to encode the text. First the suspect image is inputted and converted to binary. The first 32 bytes are analyzed to look for the amount of characters in the message. That value is then used to loop over the appropriate remainder of the image, shifting the indices of each byte left and using the AND binary operator to eliminate all bits besides the least significant one. Those bits are then collected together and converted to the hidden ASCII message. If the algorithms used cant be detected by commercial software, it will ensure the validity of the program and will demonstrate how viable the program can be if implemented in a real-world situation or within the intelligence community.



4 Results

An advanced implementation of steganography that passes at least a visual inspection will be able to be used and will not incite the suspicions of someone intercepting the message. Because of the GUI, the program will be incredibly easy to use and will display the image and what is being encoded. Then, if the decoding feature is used, it will display what is purportedly being hidden. In order to demonstrate that the program worked, the compromised image will be compared to the original. The the binary code for each will be compared and

will yield the discrepancy due to the usage of steganography. So far the GUI looks like the following:



A comprehensive GUI enables the user to encode text messages or images into a carrier image. The program will also enable a user to upload a suspected compromised image to be tested for its legitimacy. Because LSB steganography is the method of choice for professional software, it is also the easiest to detect because most software is made to be able to detect its own products. As a result, further work with mixing cryptography and steganography will have to be done to improve the program's resiliency. The advantage of using LSB encoding is its high storage capacity. A file must have eight bytes for each encoded character, which is relatively low compared to other alternatives. The initial research consisted of learning the binary structure of the various image formats and how to manipulate binary code. In addition, pixel structures and their manipulation was also researched in order to hide communications in their unused space. There are also numerous available steganography techniques, but few can encode as much information as LSB encoding. Some techniques involve digital watermarks, random insertion of fake messages, and password protection.

4.1 Experimentation

The runtime of the encoding and decoding algorithms was also tested with a variety of image inputs. The following figures show the trend:

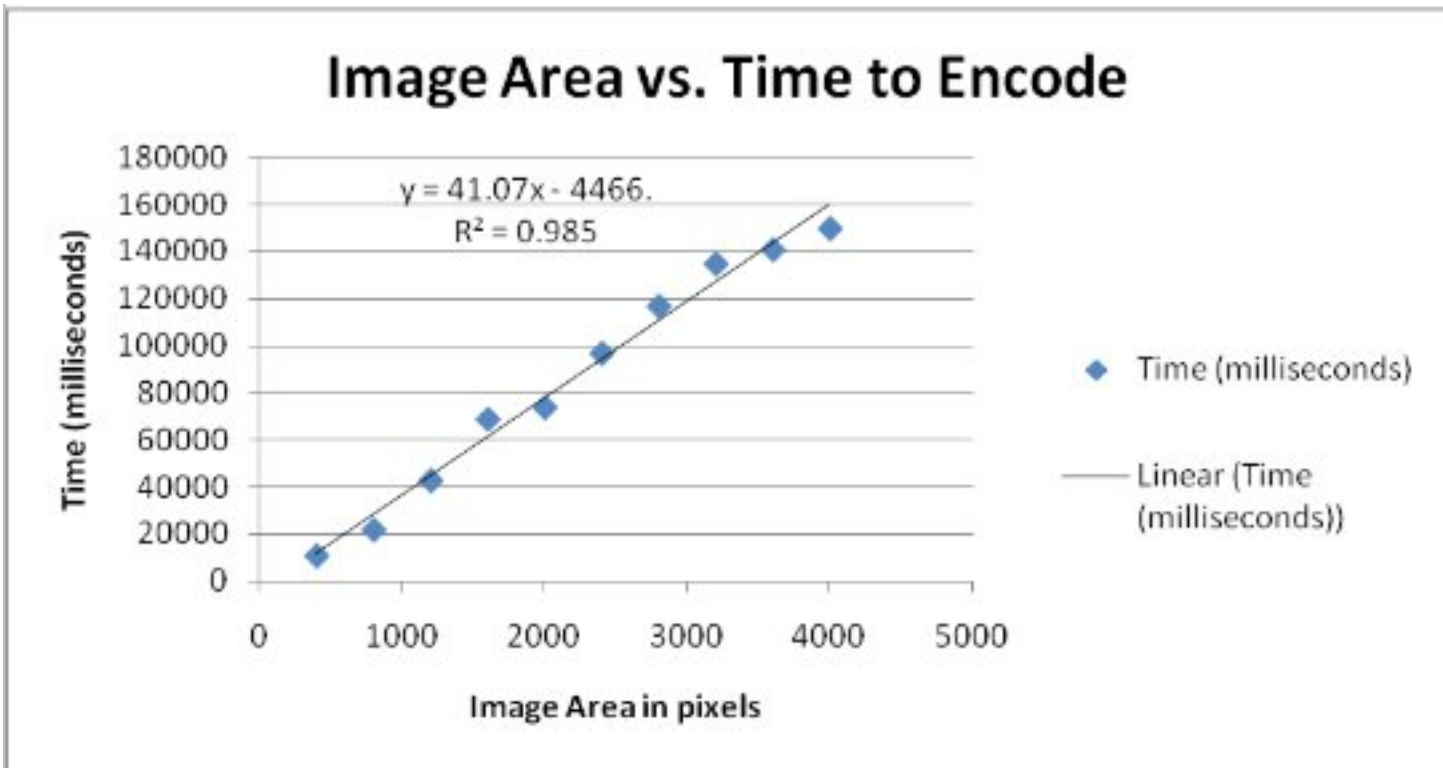


Image Area vs. Time to Encode
y = 41.07x - 4466.
R² = 0.985
Time (milliseconds)
Linear (Time (milliseconds))
Time (milliseconds)
Image Area in pixels
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1000
2000
3000
4000
5000

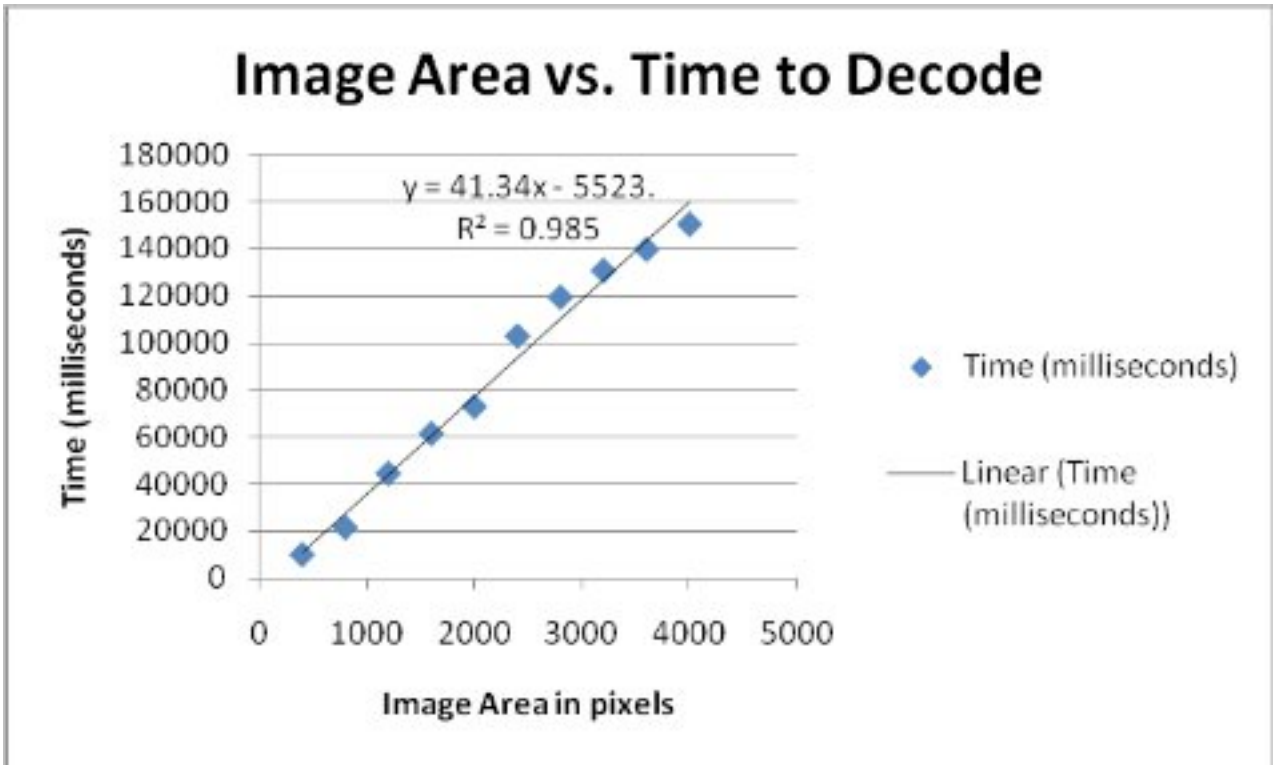


Image Area vs. Time to Decode
y = 41.34x - 5523.
R² = 0.985
Time (milliseconds)
Linear (Time (milliseconds))
Time (milliseconds)
Image Area in pixels
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1000
2000
3000
4000
5000

Its intriguing that both the encoding and decoding are linearly related, thus indicating an overall big O of O(n), but that the intercepts (not the slopes) are different, which accounts for the decoding times steepness. This might be explained by variations in the computer environment or simply the availability of system resources. The difference in the encoding and decoding doesnt appear to be statistically significant though. The following is the image size vs. encoding time graph that was generated last quarter for comparison.

5 Conclusion

A project utilizing steganography was previously done in this Senior Technology Lab in 2007 with .wav audio files. In contrast to the image steganography I am implementing, a future researcher can further investigate detection methods or steganography in .exe or other formats with white space. Similar to image steganography, far more information can be hidden in videos, which are just a compination of a series of images and sound. Videos can be effective because they vary in file size so a large file size would not be suspected. The only disadvantage is in fact the video size, which prevents the easy transmission of the message. The current program's features rival those of a commercial steganography and steganalysis program.

References

[1] Anderson "On the Limits of Steganography", 1998

[2] Bahar, H.B. "Image Steganography, a New Approach for Transferring Security Information", 2008

[3] Dabeer, Onkar "Detection of Hiding in the Least Significant Bit", 2004

[4] Dunbar, Bret "Steganographic Techniques and their use in an Open-Systems Environment", 2002

[5] Fridrich, Jessica "Practical Steganalysis of Digital Images State of the Art", 2002

[6] Hopper, Nicholas "Probably Secure Steganography", 2002

[7] Thampi, Sabu "Information Hiding Techniques: A Tutorial Review", 2004