Stegano
Research Manual

By

Sean Reidy

C00227196

Institute of Technology Carlow

November 2020

Contents

Abstract

Steganography works by replacing bits of useless or unused data with bits of different, invisible information.

Stegano is a tool that users can use to conceal a message within an image or audio file. Stegano also allows the user to encrypt the message before embedding it to enhance security measures. Stegano will be able to detect changes made to images by comparing MD5 hash values.

This tool will have 5 core functions:

1. Embed a secret message
2. Extracting a secret message
3. Encrypting a secret message prior to embedding
4. Decrypting a secret message after extracting
5. The ability to compare hash values of image files

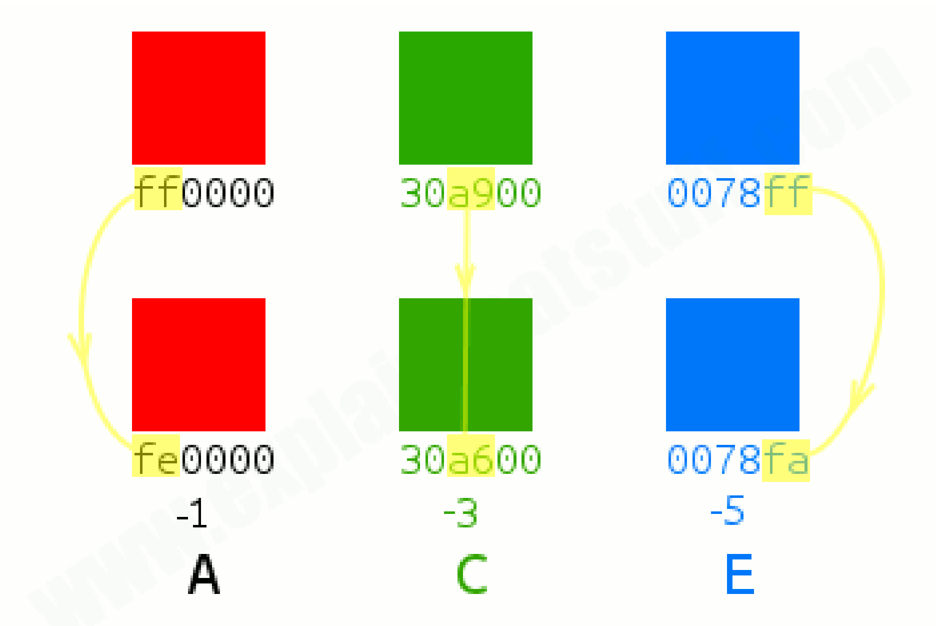
Introduction

Steganography is still widely used in today's world. The purpose of steganography is covert communication to conceal a message from a third party. In comparison to this we see that cryptography relies on the art of secret writing which makes the message unreadable by a third party, but this doesn't hide the existence of the secret message and a third party is still able to see that a message was sent from point A to point B.

Embed a secret message

To embed a secret message we will be using the Least Significant Bits approach.[1] Indexed bitmap images are made up of 2 matrices. We will be focusing on the CData matrix which is made up of unsigned 8 bit integers. We will be changing the lowest bits from the CData to hide our secret message. In every single byte(8 bits) we look to change the last bit. If we attempt to change anymore bits the image may become visibly different. We will need to encode the secret message prior to embedding it. We will then proceed to create a matrix of binary numbers which we will use in order to replace an element in the cover image.

In steganography, a common technique for embedding a secret message into an image is the Least Significant bit technique or LSB for short.[2] However, with this technique any form of image manipulation can ruin the secret message. This technique embeds the secret message by replacing some of the information in the pixels from the cover image with information from the secret message.



[7]

First step is to convert the secret message data from a decimal format to binary format. Then convert the cover image from decimal to binary too, this creates a matrix with 8 columns, one for each byte. Once this is done, break each byte down into bits.

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
10010010	11100011	01100111	00110101	11001100	00110011	1010101010	01111001

[10010010] becomes -> 1 0 0 1 0 0 1 0 and we replace the last bit, this byte now becomes -> 1 0 0 1 0 0 1 1

Now that we have each byte from the image broken down to bits we can go ahead and replace the last bit in each column with the first bit from our secret message. We shall then repeat the process for all the bytes of the cover image.

Extracting a secret message

To extract the secret message from our stego image we will simply use the reverse process of embedding an image where a user would break down the image and extract the LSB bits and convert from binary to decimal to produce our secret text.

Encrypting a secret message prior to embedding

We can choose to encrypt the secret message prior to embedding it into a cover image. This should help to raise the security of the message. However, encrypting the message will require a key which will need to be sent via "out of bounds" method. To encrypt the secret message, we will be using Advanced Encryption Standard (AES).[3] With AES our secret message will be divided into a 4x4 column of 16 bytes because we will be using AES that uses 128-bits.

The steps involved are below:

1. Produce round keys from the cipher key
2. Initialise the state array with the block data
3. Add the first-round key to the starting state array
4. Complete nine rounds of state manipulation
5. Perform a 10th and final round of state manipulation
6. The state array will contain the encrypted text.

Decrypting a secret image after extraction

After extracting the secret message, the user will need to decrypt it using the key, which will be the string which was sent via an out of bounds method. We then use the above operations in reverse order with their inverse.

The ability to detect steganography using benfords law

The tool will be able to detect if an image is using steganography by using Benfords Law.[4] Benford's law can detect the probabilities of highly likely or highly unlikely frequencies of numbers in datasets. The algorithm will be able to detect if a file is a suspicious stego-carrier by analysing the bits in each byte to. Using a generalized form of benfords law, we should be able to perform a statistical attack on an image to tell whether or not it is a stego carrier.

The general form for Benfords law is:

$$P(d) = \log_b(d+1) - \log_b(d) = \log_b\left(1 + \frac{1}{d}\right)$$ [5]

Using an image as a key with AES

To generate a key using an image we need 16 points from an image where each point will be one byte of the key. Once the 16 bytes have been generated, we must then watermark the secret key in the image so it can be sent along. We can use the lower bits of the image for this.

Solution

The Stegano tool will help users achieve easy covert communication by allowing them to embed a secret message that can be encrypted or not into an image. The tool will be coded using Eclipse and an app will be created using Android studio/Xamarin. These services in a single place should allow a user to achieve greater privacy when communicating.

The development of this project will utilise an Agile approach to help breakdown the different functions and to make sure that the single developer has working pieces before progressing onto the next stage. Using agile should allow the developer to achieve greater results and help speed up the time it takes to achieve an end product.

Similar Applications

Xiao Steganography:

This tool is free to use and can be used to hide secret messages or files in wav or BMP images. It features an easy to use gui and also supports encryption using a variety of algorithms. For a person to read a hidden message that was embedded using this tool, would need this tool to do so. This tool doesn’t offer any way of detecting steganography.

Image Steganography:

This is yet another free tool to hide a secret message in an image. It doesn't support the use of a wav file for embedding. It doesn't support encryption either. It provides a gui that allows the user to embed and extract secret messages.

Steghide:

This steganography tool is open-source and lets you embed your secret message in audio and image files. However, there is no gui provided and a user must use command-line to operate this tool. This tool only runs on Windows 32-bit systems too.

	Supports Encryption	Image and Wav files	Detects steganorgraphy	Provides a GUI
Xiao Steganography	Y	Y	N	Y
Image Steganography	N	Image only	N	Y
Steghide	N	Y	N	N

Tools used to detect Steganography.

Through-out my research I could not find a tool that was used to detect steganography by using Benfords law. Many of the forensic tools detect steganography by comparing MD5 hashes. There are other ways of detecting steganography too.[8]

Encase:

This computer forensic tool is widely used by investigators because of its powerful capabilities. It features an easy to use GUI which allows an investigator to process large volumes of computer evidence. To identify steganography files, investigators must import or build a library of hash-sets. These hash-sets are whats used to identify any steganography files on a system.

File Compression:

Steganography can be identified by using file compression. If the original image is found, one can simply compress the original and the carrier file and compare the findings. By compressing the two files one can notice that the original file looks better than the carrier file. This way of detection is also known as the known-cover attack.

ILook:

ILook is another forensic analysis tool that can be used to detect steganography by comparing file properties and MD5 hashes to determine an outcome. This tool is a popular tool amongst investigators.

Software Development Tools

Android Studio

Android studio would be my go-to platform for designing an app. During my work placement in third I created an app using this so I'm familiar with it. Not to mention that I have studied java for the majority of my 4-year course.

Eclipse

Eclipse would be a very close second choice to me. I spent the whole of 2nd year using this IDE and became familiar with it. However, Android studio is easier to use and some features are not included in Eclipse, for example, in android studio you can use pre-made backgrounds for the app design.

Xamarin

This tool would be the least familiar to me and one that would require more learning and research than the others. Xamarin is open-sourced and based off of the .NET framework. It also uses C# to create a mobile app. To create a mobile app I would have to pair this up with the likes of Visual Studio.

Overall Android Studio would be my preferred choice to use in the development of this app.

References

1	Fridrich, J., Goljan, M. and Du, R., 2001. Detecting LSB steganography in color, and gray-scale images. *IEEE multimedia*, *8*(4), pp.22-28.
2	Neeta, D., Snehal, K. and Jacobs, D., 2006, December. Implementation of LSB steganography and its evaluation for various bits. In *2006 1st International Conference on Digital Information Management* (pp. 173-178). IEEE.
3	GNDU RC, J., 2015. Dual layer security of data using LSB image steganography method and AES encryption algorithm. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *8*(5), pp.259-266.
4	Bonettini, N., Bestagini, P., Milani, S. and Tubaro, S., 2020. On the use of Benford's law to detect GAN-generated images. *arXiv preprint arXiv:2004.07682*.
5	En.wikipedia.org. 2020. *Benford's Law*. [online] Available at: <https://en.wikipedia.org/wiki/Benford%27s_law#Benford's_law_in_other_bases>
6	GreatLearning. 2020. *Image Steganography Explained | What Is Image Steganography?*. [online] Available at: <https://www.mygreatlearning.com/blog/image-steganography-explained/>
7	Explainstuff.com. 2020. [online] Available at: <https://www.explainthatstuff.com/encryption.html> [Accessed 10 November 2020].
8	Sans.org. 2020. [online] Available at: <https://www.sans.org/reading-room/whitepapers/stenganography/steganalysis-detecting-hidden-information-computer-forensic-analysis-1014> [Accessed 09 November 2020].