Steganography in digital media

Pricope Stefan-Cristian
psir2384@scs.ubbcluj.ro

Suciu Mihai
mihai-suciu@cs.ubbcluj.ro

Abstract

Steganography is the science of concealing a piece of information within another piece of information without affecting the latter in a noticeable way and therefore alerting any intruders of the existence of the former. This paper presents both existing and new ways of embedding computer files and data into different digital multimedia formats as covert as possible while still allowing the encoded information to be retrieved at a later time without any significant losses.

Contents

Chapter 1

Introduction to computer steganography

The practice of hiding and securing information and messages between different parties has played a major part alongside human history, especially during times of war when it was vital that the enemy didn't intercept the strategies and even if they did, they would have no idea what to do with them and would have to dedicate a lot of time, money and personnel to decode the communications. Two of the most famous manifestations of this practice are cryptography and steganography[1].

Trying to differentiate between cryptography and steganography is not difficult, the only thing they have in common is their end goal - making sure that a piece of information sent from one place to another is secured and that only the right recipient will be able to read and understand the received information. The difference lies in the methods they use and the time it takes to reach that goal.

Cryptography focuses on altering the information itself, encrypting it using a key only known by the receiver and sender[1], making it harder for any possible meddlers to alter the meaning of the message or even just understand it. Steganography is more concerned on hiding the fact that there even is any information being transmitted usually by embedding it in something else (hereby referred to as a cover), thus if any interlopers were to actually look at the cover, they wouldn't even be aware of the fact that it contained secrets.

In other words, both methods are meant to be used over an open and unsafe environment, and while cryptography tries to hide the contents of the message but not the fact that there is a message being sent, steganography tries to hide the communication altogether. The main advantage of these 2 methods is that they are not exclusive, they can be used together for maximum safety of the information.[2]

With the evolution of the Internet and the increasing usage of personal computers in day to day activities we needed to secure the data sent over the network between the users. The efforts were lead by cryptographers who developed thousands of algorithms for encrypting the information and the very few remarkable ones are still being used[3]. But that doesn't mean steganography was left behind, researchers developed plenty of new algorithms for hiding information using digital files as a cover and the Internet as the transmission environment.

In theory all types of digital files can be used as cover - shared libraries and executables can be edited to include the hidden data and then be recompiled/rebuilt, text files and documents could be modified to enclose the information in secret places/paragraphs, and multimedia such as images, music and video files can be changed to carry the digital data into their metadata, pixels, audio frequencies, motion vectors and much more. This paper will go into details about some of the most popular formats used for multimedia files and showcase their internal structure and document techniques used in steganography.

[1]This is only true in symmetric cryptography and is a gross oversimplification of cryptography as a science, but it is just meant to get a point across and help differentiate between the 2.

[2]The speed of encoding/decoding the information is greatly decreased when these 2 are combined which is the reason that nobody merges them, preferring to juse use encryption to keep the information safe.

[3]The reason cryptography is in the spotlight is because it is much faster, mathematically proven, and it hides all the data without leaving anything in plain sight(like steganography does with the cover).

Chapter 2

Image file formats and steganography techniques

2.1 Introduction

An image is a two-dimensional representation depicting any possible subject conceivable by human imagination, captured using an optical device (such as a camera or a telescope) or a natural object (human eyes). The image can then be rendered and displayed for other people to see either manually (by painting, carving etc.) or automatically (by using a computer). In this chapter we will focus on images captured using digital optical devices that are rendered automatically. The correct term for them is digital images, but throughout the rest of the paper they will be reffered to as images for convenience.

Figure 2.1: Lenna - Classic example of a digital image

Computers are programmed to do operations in a clear sequential way and this rule doesn't change when working with pictures. In order for a computer to be able to render an image, it needs to know some general metadata information about the photo, such as the width and height, as well as the data bytes of the image. These bytes are the actual representation of the picture which compose the two-dimensional pixel map[1]. A pixel is the smallest unit that a computer monitor can read and display. The pixel color is the result of merging the different color channels which compose the picture (such as RGB, YUV, YCbCr etc.). Here is an example of the entire process - let's assume that from the image data bytes the first 3 bytes have the decimal values 20, 127, 250 and that it uses the RGB color model. This means that when the computer will have to render the image, the first pixel will have the red component equal to 20 (0x14), the green equal to 127 (0x7F), and the blue equal to 250 (0xFA), in what will finally be interpreted as #147FFA by the monitor (variation of light blue).

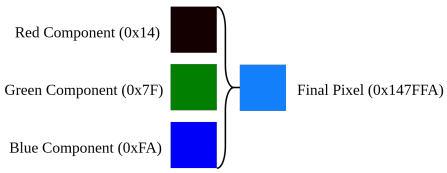

Figure 2.2: How 3 colors channels build the pixel

With all of this information in mind, we can now procede to discussing the different steganography methods possible when working with images as cover.

[1]This is true for a lossless format, where each pixel is stored in memory. It is not exactly the case for lossy formats such as JPEG where the image goes through processing before being rendered. More information later in the chapter.

2.2 General and particular algorithms in image steganography

2.2.1 Least Significant Bit (LSB)

Least Significant Bit or LSB is by far the most used method when talking about any type of steganography. Given that the smallest unit a computer can understand and process is usually a byte, altering only the least significant bit will not change the transmitted information in a noticeable way to any external parties. It is much easier to showcase what a byte contains and what the LSB change implies and how it works. A byte contains 8 bits, so this means that the values a byte can take range anywhere from 0 to 255 (inclusive)[2]. Let's assume that we have an array of 4 random values in consecutive memory : 217, 127, 100, 62 (all values are in decimal), each stored on exactly one byte, and that we want to hide our grade in Numerical Analysis from our parents (in this case a 3) using a LSB substitution. The process would be something like this :

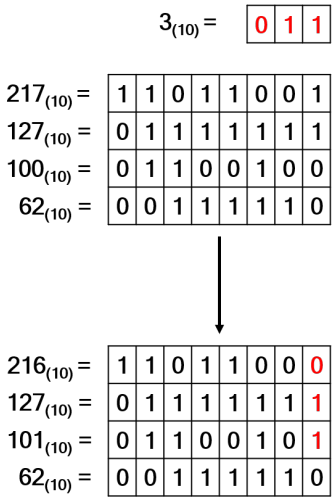

Figure 2.3: How the Least Significant Bit change works

As we can see from Figure 2.3, we have succesfully altered the least significant bit of the first 3 bytes of the stream in order to hide our grade : 217 became 216 when we changed the last bit from 1 to 0, 127 was unchanged because it already had the last bit set, and the third byte became 101 after toggling the final bit. Furthermore, the rest of the stream (the fourth byte, 62) was not affected because we already hid the entirety of our secret message. While this is great because we only hide exactly as much as we need and not a byte more, we have a high risk of corrupting the hidden message in case our cover image gets compressed or loses even a single byte when sent over a network. Basically, we are trading data redundancy in order to get simplicity and efficiency.

[2]This is the case for unsigned bytes, but given that we are talking about a method that only deals with the least significant bit, we can safely ignore the most significant bit, also known as the sign bit.

2.2.2 Metadata encoding

test2

2.2.3 Unused bytes

2.3 BitMap Picture (BMP)

2.4 Portable Network Graphics (PNG)

2.5 Joint Photographic Experts Group (JPEG)

Chapter 3

Audio file formats and steganography techniques

3.1 Introduction

3.2 Techniques used in audio steganography

3.2.1 Least Significant Bit (LSB)

3.2.2 Metadata encoding

3.2.3 Embedding within certain frequencies

3.3 The MPEG-1/2 Audio Layer III (MP3)

3.4 Waveform Audio (WAV)

Chapter 4

Video file formats and steganography techniques

4.1 Introduction

4.2 General and particular algorithms in video steganography

4.2.1 Image and audio steganography combinations

4.2.2 Motion Vectors

4.3 MPEG-4 Part 14 (MP4)

4.4 Audio Video Interleave (AVI)

Chapter 5

Conclusion

Chapter 6

Bibliography

Bibliography

[1] Petitcolas FAP, Anderson RJ and Kuhn MG. *Information Hiding - A Survey*. Proceedings of the IEEE, special issue on protection of multimedia content, 1999.

[2] Merriam-Webster dictionary.
`https://www.merriam-webster.com/dictionary/steganography`

[3] Johnson Neil and Jajodia Susil. *Exploring Steganography : Seeing the Unseen*. Los Alamitos, IEEE Computer Society, 1998.

[4] T. Morkel, J.H.P. Eloff and M.S. Olivier. *An overview of image steganography*. University of Pretoria, ICSA Research Group, 2005.

[5] Niels Provos and Peter Honeyman. *Hide and Seek: An Introduction to Steganography*. University of Michigan, IEEE Computer Society, 2003