

Robust Steganography for Online Social Networks using the Tagging Steganography Scheme

by

Matthew Mills

Submitted in partial fulfilment of the requirements for the degree
Bachelor of Science (Computer Science) [Hons]
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria

November 2014

FORM A

UNIVERSITY OF PRETORIA

FACULTY EBIT
DEPARTMENT Computer Science

The Department of Computer Science places specific emphasis on integrity and ethical behaviour with regard to the preparation of all written work to be submitted for academic evaluation.

Although academic personnel will provide you with information regarding reference techniques as well as ways to avoid plagiarism, you also have a responsibility to fulfil in this regard. Should you at any time feel unsure about the requirements, you must consult the lecturer concerned before you submit any written work.

You are guilty of plagiarism when you extract information from a book, article or web page without acknowledging the source and pretend that it is your own work. In truth, you are stealing someone else's property. This doesn't only apply to cases where you quote verbatim, but also when you present someone else's work in a somewhat amended format (paraphrase), or even when you use someone else's deliberation without the necessary acknowledgement. You are not allowed to use another student's previous work. You are furthermore not allowed to let anyone copy or use your work with the intention of presenting it as his/her own.

Students who are guilty of plagiarism will forfeit all credit for the work concerned. In addition, the matter can also be referred to the Committee for Discipline (Students) for a ruling to be made. Plagiarism is considered a serious violation of the University's regulations and may lead to suspension from the University.

For the period that you are a student at the Department of Computer Science, the under-mentioned declaration must accompany all written work to be submitted. No written work will be accepted unless the declaration has been completed and attached.

I (full names) Matthew Mills
Student number 11086697
Subject of the work Robust Steganography for Online Social Networks using the Tagging Steganography Scheme

Declaration

1. I understand what plagiarism entails and am aware of the University's policy in this regard.
2. I declare that this dissertation (e.g. essay, report, project, assignment, dissertation, thesis etc) is my own, original work. Where someone else's work was used (whether from a printed source, the internet or any other source) due acknowledgement was given and reference was made according to departmental requirements.
3. I did not make use of another student's previous work and submitted it as my own.
4. I did not allow and will not allow anyone to copy my work with the intention of presenting it as his or her own work.

Signature 

Robust Steganography for Online Social Networks using the Tagging Steganography Scheme

by

Matthew Mills

Email: contact.matthewmills@gmail.com

Abstract

On the internet today, there are a number of websites offering the ability to store and share images. These Online Social Networks (OSNs) provide an inconspicuous and ideal platform to communicate covertly using image steganographic techniques due to the millions of images that are shared publically on them on a daily basis. The issue for steganographers is that generally OSN's manipulate the images uploaded to them in a number of ways such as by resizing, compressing and embedding metadata. This makes it extremely challenging to use popular steganographic techniques on these websites as these modifications to the image usually destroy any hidden message within.

This work explores the adaptation of a robust watermarking scheme called Tagging, in order to enable images containing hidden data to survive modifications made by OSN's.

Keywords: information hiding, steganography, watermarking, social network, tagging, OSN, image sharing

Supervisor: Ms. T. Morkel

Department: Department of Computer Science

Degree: Bachelor of Science (Honours)

Contents

List of Figures	i
List of Tables	ii
Chapter 1 - Introduction	1
1.1 Introduction	1
1.2 Research Objectives	3
1.3 Research Methodology	3
1.4 Contributions	3
1.5 Chapter Layout	4
Chapter 2 – Steganography	5
2.1 Introduction	5
2.2 Overview of steganography	5
2.3 Uses of steganography	7
2.4. Image steganography	8
2.4.1 Least Significant Bit (LSB) steganography	8
2.4.2 JPEG steganography	10
2.5 Applicability of current techniques for OSN's	14
2.6 Conclusion	16
Chapter 3 – Digital Watermarking	17
3.1 Introduction	17
3.2 Overview of Watermarking	17
3.3 Types of Watermarking Schemes	18
3.4 Robust Watermarking: Tagging	19
3.5 Applicability of Tagging for OSN's	22
3.6 Conclusion	23
Chapter 4 – The Tagging Steganography Scheme	24
4.1 Introduction	24
4.2 Requirements for successful steganography on OSN's	24
4.3 Technical details of the implementation of the Tagging Steganography Scheme	26
4.3.1 Choosing an OSN	26
4.3.2 Ambiguous Terminology	26

4.3.3 Tagging Considerations for OSN's.....	27
4.3.4 Embedding of Tags.....	27
4.3.5 Message Extraction.....	32
4.4 Conclusion.....	33
Chapter 5 – Experimental Results.....	34
5.1 Introduction	34
5.2 Experimental Results.....	34
5.3 Parameters.....	37
5.3.1 Tag Brightness Modulation	37
5.3.2 Block Size.....	38
5.3.3 Thresholds.....	40
5.4 Analysis of the Tagging Steganography Scheme.....	41
5.5 Conclusion.....	42
Chapter 6 – Conclusion.....	43
6.1 Summary	43
6.2 Future Research	43
References.....	44

List of Figures

2.1 The Steganographic operation	6
2.2 A Pixel's RGB components and byte values	9
2.3 Embedding text in LSB's	10
2.4 Quantized DCT coefficients and their equivalent binary representation	13
2.5 DCT coefficients in a normal image and in an image containing hidden information using JSteg	13
3.1 Locations are analysed and those that are suitable are chosen for tagging	19
3.2 A tagged image with tags exaggerated (20% brightness modulation)	20
3.3 A tagged image (2% brightness modulation)	20
4.1 The design trade-offs of a steganographic system	25
4.2 The magic triangle for (a) LSB Steganography and (b) watermarking	25
4.3 A text string is converted to its byte equivalent representation	28
4.4 Tagging locations are identified using thresholds	30
4.5 Tagging locations are identified using best-x	30
4.6 Extraction Process for Tagging on OSN's	32
5.1 A message is embedded using the Tagging Steganography Scheme	35
5.2 The image before tagging (A), and after tagging (B)	35
5.3 The stego-image is posted to Facebook and available for download	36
5.4 The image is analysed and the message is successfully retrieved	36
5.5 Brightness modulation of tags	38
5.6 JPEG Compression degrades image information	39
5.7 Suitable and unsuitable tagging locations as determined by thresholds	40

List of Tables

3.1 Summary of the differences between Steganography and Watermarking	18
5.1 Recovery rate of various block sizes	39
5.2 Histograms and variance of different blocks	40

Chapter 1 - Introduction

1.1 Introduction

In the simplest sense, steganography is the *art of hiding information inside other information* [1]. The desire to hide information is not new, with the earliest known uses traceable to Ancient Greece [2]. In the modern era, the goal to communicate covertly has remained, but the tools at hand have changed significantly. A variety of steganography techniques have been developed for a multitude of different carrier formats and digital images, audio files and video clips can all be used as carriers for hidden information. Images remain the most popular cover medium due to their frequency on the internet [3].

Simply by using a steganography algorithm, you are not guaranteed to hide the presence of communication altogether. A back-and-forth email conversation consisting mainly of large images will certainly appear suspicious to even a casual observer. Instead, one has to use steganography carefully and selectively in order to avoid detection [1]. It is for this reason that it can be useful to use steganography over a public communication channel, such as a social network.

Online social networks (OSN's) are dedicated websites that enable users to communicate with each other by posting information, comments, messages, images, etc. [4]. Websites such as Facebook, Twitter, Instagram and many others have become extremely popular in recent years. According to research, social media has overtaken email as the most popular online activity – so much so that social networking accounts for nearly 10% of all time spent on the internet [5]. Users post over 350 million photos on Facebook per day, making it the largest photo sharing website in the world [6]. Clearly OSN's offer a potential public platform for secret communication – a handful of images containing hidden data would not be likely to stand out

against the millions of normal images. An issue steganographers face is that OSN's tend to modify the images uploaded to them, for example, resizing or compressing the images or by embedding metadata in them. While these modifications are not likely to be visually perceptible, typical steganography algorithms are not usually robust enough to handle these modifications, making it very difficult to use them on OSN's.

Watermarking is a related technique that is often used as a tool to protect intellectual property. While similar to steganography in the sense that it also embeds data into a cover medium, the goals for watermarking differ from those for steganography [7]. Watermarks typically only need to embed a small amount of data (such as information about the origin/destination of the image), but needs to be robust enough to withstand attacks in order to protect this data.

Steganography and watermarking can thus be combined to provide a possible solution for an algorithm that can be successfully be used by two parties to communicate covertly using images on an OSN. The algorithm needs to be imperceptible – one should not be able to notice a difference between an original image and an image with embedded data. The algorithm should also provide sufficient robustness to withstand modifications made by OSN's. Thirdly, the algorithm should have as large a storage capacity for hidden data as possible.

This dissertation proposes using a watermarking technique known as Tagging [8] in the implementation of a steganography scheme to be used for the hiding of data within images on OSN's while providing imperceptibility, robustness and a high storage capacity.

The rest of this chapter is outlined as follows: Section 1.2 lists the main objectives of this dissertation, section 1.3 describes the research methodology for the dissertation, section 1.4 summarises the original contribution of this work and section 1.5 outlines the structure of the rest of this dissertation.

1.2 Research Objectives

Simplified, the problem to be addressed in this dissertation is to find a way to be able to use steganography on OSN's. Current popular steganography algorithms lack the robustness needed to withstand the modifications applied by OSN's. It is therefore proposed to adapt the Tagging watermarking algorithm for this purpose. The primary objectives are summarized as follows:

- To provide an overview of popular steganographic algorithms, namely LSB Steganography for bitmap images and JSteg for JPEG images and explain why they are not applicable for use on social networks.
- To provide an overview of the watermarking scheme, Tagging and demonstrate how it can be applied to reliably embed and retrieve information on social networks.
- To provide results of tests carried out on Facebook using the proposed scheme.

1.3 Research Methodology

In order to solve the problem at hand, an extensive literature study was done in order to gain knowledge of current techniques. Secondly, a practical implementation of a solution was created and tested on an OSN before examining experimental results.

1.4 Contributions

This dissertation contributes to the field by introducing an application of the Tagging watermarking scheme [8] that can be used to reliably embed and retrieve data in images on OSN's, despite the modifications these website make to them.

1.5 Chapter Layout

The remainder of the dissertation is organised as follows:

- **Chapter 2** discusses popular steganography algorithms and explains why they are not suitable for use on OSN's.
- **Chapter 3** looks at watermarking and how it differs from steganography, focusing specifically on the Tagging watermarking scheme as the basis for a solution.
- **Chapter 4** is dedicated to discussing in detail how Tagging can be used as a robust steganography scheme for OSN's.
- **Chapter 5** looks at tests and experimentation that were carried out on Facebook using the proposed scheme.
- **Chapter 6** presents ideas for future research and concludes the dissertation.

Chapter 2 – Steganography

2.1 Introduction

This chapter aims to provide background information on steganography – one of the fundamental techniques that is used in the proposed solution.

Firstly, two current steganography algorithms are analysed to discover why they are not viable for use on OSN's. In order to do this, background information on steganography and the fundamentals of these algorithms are described.

This chapter gives an overview of steganography and the current popular algorithms in Section 2.2, section 2.3 discusses the various uses of steganography, section 2.4 discusses current algorithms used for steganography while section 2.5 discusses why these techniques are not suitable for use on OSN's. Finally, a conclusion is presented in section 2.6.

2.2 Overview of steganography

The desire to hide secret communication in plain sight is not new; in fact the ancient Greeks had a number of clever techniques for communicating covertly [2]. For example, they would shave the hair off a messenger and tattoo an important message on their head. Once the hair had grown back, the message would remain undetected until their head was shaved again. Another method involved writing text on wax-covered tablets. One could scrape off the wax, write a message underneath and then cover it up with wax again to make it seem unused [9]. More recently, during World War II, invisible ink played an important role in discrete communication – a seemingly-innocent message could carry a far more important message between the lines [10].

Modern steganography can be succinctly described by the *prisoner's problem* proposed by Simmons [11]. In this scenario, Alice and Bob are two inmates who are imprisoned separately but wish to communicate in order to hatch an escape plan. Any communication between them

has to go through a warden, Willie, who will put them in solitary confinement and forbid any further messaging if he suspects they are communicating secretly [12]. Alice and Bob therefore have to find a way to conceal their plans by hiding it in innocuous-looking coverttext in order to evade detection.

The objective of digital steganography is thus to hide a secret message inside a cover-media in such a way that others cannot discern the presence of the hidden message [13]. A secret message can range from plaintext, cyphertext, image or anything else that can be represented as a bit stream [14]. Digital images, videos, sound files and other computer files contain perceptually redundant or irrelevant information can be used as cover-media [7].

Hiding information in media requires the following elements [15]:

- **Cover media** (C) that will hold the hidden data
- Secret **message** (M), that can take the form of any sort of data
- The **stego function** (F) used to embed the message and its **inverse** (F^{-1}), used to retrieve the embedded message
- An optional password or **stego-key** (K)
- The stego function produces the **stego object** (S)

A diagram representing these elements is depicted below:

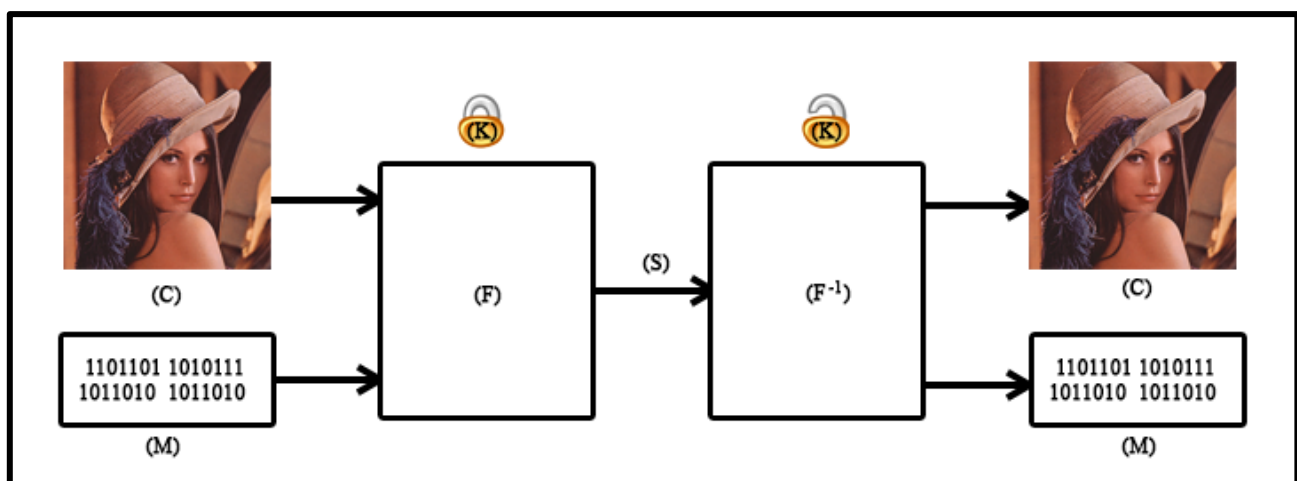


Figure 2.1: The Steganographic operation

A sender selects cover media of their choice (C) and converts their message (M) into a bitstream. They then apply the stego function (F) which embeds the bitstream within redundant information in the cover object, producing the stego object (S). The stego object is typically then sent over some form of communication channel. In our case it will be via an OSN, but there are a variety of options such as via email, file stream or other means. There is also the possibility to use a shared password or key (K) for extra strength which would be required for decrypting of the message using the inverse of the stego function (F^{-1}). Once applied, this inverse function reveals the embedded information to the recipient.

This model is applicable for most forms of steganography. Steganography typically takes advantage of redundancy – bits of data that can be altered without being noticed. Where redundancy can be found in cover media, steganography can take place [16].

2.3 Uses of steganography

There are many cases where steganography can prove useful. As a technique it can be used to conceal important personal details and when paired with existing communication methods, it can be used to carry out secret exchanges of information [17].

Currently both cryptography and steganography serve as important tools in securing our data. The purpose of steganography is to hide the very presence of communication as opposed to cryptography whose goal is to make communication unintelligible to those who do not possess the right keys [16]. Cryptography has known shortcomings – specifically, the presence of an encrypted message itself may be suspicious [18]. It can be argued that if one goes to the trouble of encrypting something then surely it must be important. Steganography addresses this by concealing the fact that a message is being sent altogether, and, if not detected, enables the sender and receiver to remain “invisible” – thus steganography potentially provides not only security, but also anonymity and privacy [6].

Steganography can be a particularly important tool when used in censored or monitored environments or in environments that place restrictions on the use of encryption [20]. There are a

number of countries where freedom of expression is severely limited, and in these cases steganography can enable one to send news and information without being censored and without the fear of messages being intercepted [20].

Another use for steganography is for storing important personal information – Information such as bank details, military secrets and internet passwords can be stored secretly in cover sources. An attacker would likely attack an encrypted file but might overlook an innocuous-looking cover image [6].

2.4. Image steganography

Images remain the most popular cover objects for steganography. The amount of redundancy created in their representation makes them ideal carriers of hidden data [14]. Images are also popular because of their frequency on the internet, be it on websites, social networks or emails – making them both accessible and unsuspecting. For the purposes of this dissertation, only image steganography is considered further. For a detailed look at steganography in other media, the reader is referred to Hayati [21].

This chapter looks at two popular steganography algorithms, Least Significant Bit (LSB) steganography for bitmap (BMP) images and JSteg for JPEG images. These algorithms were the original steganography algorithms for their respective formats and are the most simple to explain. They share the same principles as a number of their successors.

2.4.1 Least Significant Bit (LSB) steganography

LSB steganography is a simple and popular technique that works on raster images by flipping the least-significant bits of colours in each pixel as needed to match a message bit stream. While it has a number of disadvantages, it is easy to understand and implement [20]. This section starts by explaining bitmap representation before looking at how LSB steganography can be applied.

Image representation

LSB steganography takes advantage of human perception being only able to see a certain number of colours. A bitmap image uses the RGB colour model, making it essentially just a 2-dimensional grid of colours. Each colour, or pixel, is made from a combination of red, green and blue.

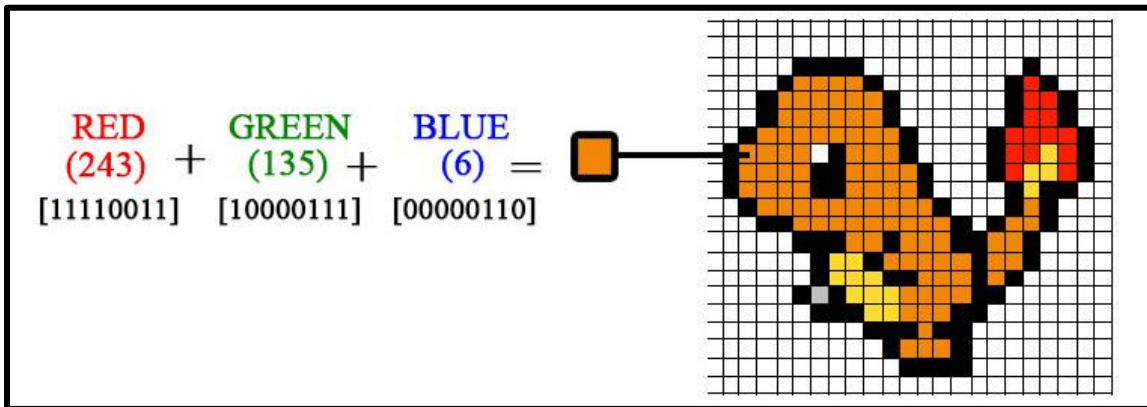


Figure 2.2: *Pixel's RGB components and byte values*

(Image credit: Nintendo)

Each red, green and blue component can store one byte worth of data – a value ranging from 0 to 255, representing the amount of that particular colour component. As can be seen in Figure 2.2, the particular orange used in the image is made from a lot of red, a bit of green and very little blue. The over 16.7 million possible colours available with this method are significantly more than what the human eye can recognise which is in the region of 10 million colours [22]. This redundancy is something that steganography can take advantage of.

Method

In order to get best results using LSB steganography, a 24-bit colour image with a lossless compression format is needed. Lossless compression allows the image to be transformed and maintain quality, opposed to a lossy compression algorithm which loses quality when transformed. Formats that would work for this purpose are BMP, GIF and PNG [23].

Using standard LSB steganography, the least significant bits of each RGB component can be modified, thus an 800x600 image would have 1 440 000 usable bits. For example, Figure 2.3

depicts how the letter 'x' can be embedded in 3 pixels by first converting plaintext into binary form and then modifying the least significant bits of each pixel. In this case, 4 bits needed to be changed (those in red) while 5 could remain the same (those in green). On average, only half of the bits will need to be changed.

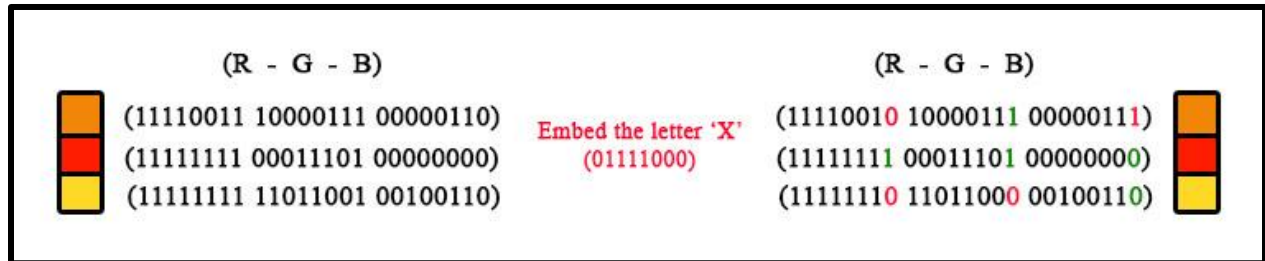


Figure 2.3: Embedding text in LSB's

As can be seen, it is not possible for the human eye to differentiate between the pre-embedding and post-embedding colours, so the message is hidden successfully.

2.4.2 JPEG steganography

JPEG is the most popular format on the internet today and for this reason has become an important research direction in the steganographic community [24]. Joint Photographic Experts Group (JPEG) is an example of a transformation domain format that uses lossy compression. These images typically have far smaller file sizes than the equivalent bitmap, making them extremely popular for use on the internet [25]. It was initially thought that performing steganography on JPEG images would not be possible, due to the compression applied – steganography embeds bits in redundant data, but if redundant is excluded during compression it was thought that steganography would be destroyed. However it has been discovered that properties of the compression algorithms can be exploited for the purposes of steganography [3]. This section starts by explaining JPEG compression in more detail so that aspects relevant to steganography can be defined.

Image representation

JPEG compression as consists of five basic steps, as described by Fridrich [7]:

1. **Colour transformation:** Colours are transformed from the RGB model to the $Y C_r C_b$ model. This model separates RGB (Red, Green, Blue) into luminance (Y) and chrominance components (C_{red} and C_{blue}). This model takes the limits of human vision into account and eliminates the redundancy in the in the RGB model [26]. The luminance component is more visually perceptible and is therefore kept intact, but chrominance components can be compressed considerably.
2. **Division into blocks and subsampling:** Luminance blocks are divided into 8×8 blocks while the chrominance signals are subsampled in order to achieve a higher compression ratio before DCT transformation. For example for each pixel, luminance might be measured, but chrominance might only be measured over every 2×2 or 4×4 block of pixels.
3. **DCT transform:** The Discrete Cosine Transform (DCT) helps separate the image into parts (spectral sub-bands) of differing importance (with respect to the image's visual quality) [27]. Pixel intensities are divided into a matrix with low frequency values appearing in the upper left and higher frequency values in the lower right. These higher values can typically be neglected with little visual distortion.
4. **Quantization:** The purpose of quantization is to enable representation of DCT coefficients using fewer bits. This is the 'lossy' part of the algorithm and is the reason why JPEG can be compressed to such small file sizes [28].
5. **Encoding and lossless compression:** The quantized DCT co-efficient are then compressed losslessly. Due to a number of zeros being introduced during quantization, Hoffman encoding in a zig-zag fashion over the matrix produces the best compression results. A header is added to the front of the resulting bitstream and stored with the '.jpeg' extension.

For a full description of the JPEG standard, the reader is referred to Pennebaker [29].

JSteg - Method

Looking at the algorithm it is clear that steganography cannot take place prior-to, or during the lossy phases, DCT and quantization. Steganography usually hides in redundant data but since these lossy phases remove redundant data, they are not suitable for steganography. The potential for steganography in JPEG presents itself after lossy compression but before lossless compression. As proposed by Upham [30], LSB embedding can be used to embed the message into the least significant bits of all non-zero DCT coefficients before applying the Huffman encoding [31].

JSteg simply embeds message bits in coefficients that are not 0 or 1. This is necessary because these values are an LSB pair (values that only differ in their LSB's) and changing these would cause a large distortion [33]. Pseudo code of the embedding process looks as follows:

Input: message_bits, cover_img

Output: stego_img

```
while bits remaining to embed
    get next DCT coefficient from cover_img
    if DCT != 0 & DCT != 1
        get next bit from message_bits
        replace DCT LSB with bit from message
    end if
    insert DCT into image
end while
```

26	3	6	2	2	1	0	0	00011010	00000011	00000110	00000010	00000010	00000001	00000000	00000000
0	2	4	1	1	0	0	0	00000000	00000010	00000100	00000001	00000001	00000000	00000000	00000000
3	1	5	1	1	0	0	0	00000011	00000001	00000101	00000001	00000001	00000000	00000000	00000000
3	1	2	1	0	0	0	0	00000011	00000001	00000010	00000001	00000000	00000000	00000000	00000000
1	0	0	0	0	0	0	0	00000001	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0	0	0	0	0	0	0	0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0	0	0	0	0	0	0	0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0	0	0	0	0	0	0	0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0	0	0	0	0	0	0	0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

Figure 2.4: Quantized DCT coefficients and their equivalent binary representation (red values represent 0's and 1's that are unsuitable for JSteg, the other values are available for LSB)

An important aspect of JSteg and JPEG steganography in general, is the lack of visual artefacts produced. Steganographic systems that modify least-significant bits of actual pixel values are often susceptible to visual attacks. In JPEG steganography however, the modifications are in the frequency domain instead of the spatial domain, making visual attacks impossible [33].

Improvements

Despite not producing visual artefacts, JSteg has been discovered to be a weak steganographic system [34]. It has been shown using statistical tests that one can easily observe how the DCT coefficients of a normal image differ from one with steganography embedded [35]. As can be seen in Figure 2.5, due to the flipping of LSB's, a peculiar histogram is typically created. Note the coefficient pairs being formed, causing a 'staircase' effect:

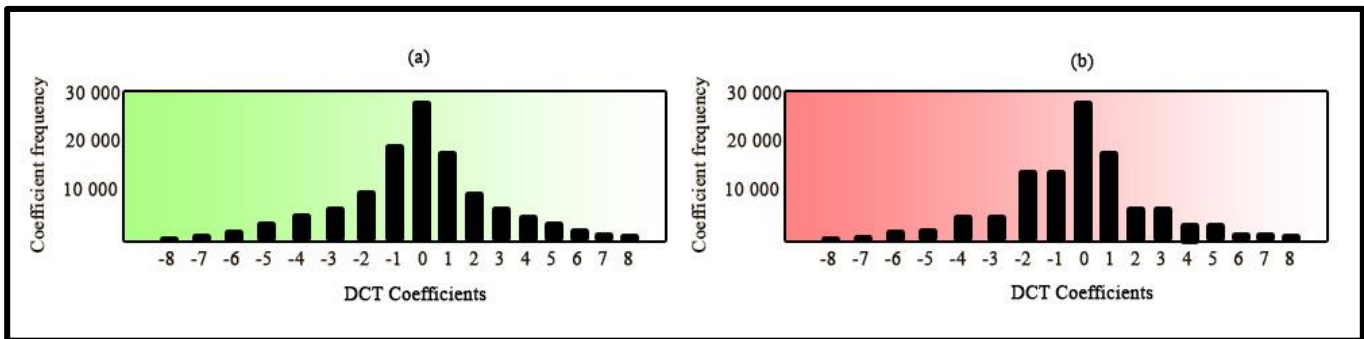


Figure 2.5: DCT coefficients in a normal image (a) and in an image containing hidden information (b) hidden using JSteg.

A few improvements to JSteg have thus been suggested:

Outguess, introduced by Provos [36], is an example of a statistically-aware steganography algorithm. Outguess embeds data in DCT coefficients that are not one or zero in the same fashion as JSteg, but afterwards corrects the histogram by modifying unused coefficients.

F5 introduced by Westfeld [37] also preserves the histogram but without the need for additional corrections. This algorithm instead decrements the coefficients absolute value to match the message bit, thus eliminating fixed pairs of DCT coefficients. Furthermore it makes use of matrix encoding to decrease the number of required changes to the cover image.

2.5 Applicability of current techniques for OSN's

As defined in the problem statement, we wish to discover why current steganographic techniques are not applicable for use on OSN's. There are a number of ways that OSN's handle media that present challenges for steganographers.

Handling of images on OSN's

Users can typically upload images of a variety of formats and sizes to OSN's but they are all typically treated in the same way – usually being converted by the OSN to JPEG format and then resized and compressed considerably. This could be done for a number of reasons:

- It allows additional applications to handle one **consistent file format and naming scheme** across the social network: Many 3rd party applications can be allowed access to the media content uploaded by users through development API's. With such a variety of different image, audio and video file types available, it makes sense that OSN's would standardise this [38].
- **Storage constraints:** With typically millions of users each typically allowed an unlimited number of posts (image/video/audio/text uploads), storage space is an understandable concern, thus storing only the smallest version of the file is sensible. [39]

- **Optimal user experience:** Users typically browse through posts, taking an interest in only a handful of these posts. If each post had to be loaded in full quality, it may take a long time for pages to load, causing a frustrating browsing experience [40].
- **Most users are not concerned** by the quality: Typically there are very few visual differences between an uncompressed file and its compressed equivalent, so much so that users are unlikely to notice or care that their files have been modified. [41]
- **To prevent steganography:** The exact extent of steganography on the internet is not known and difficult to estimate [42], however Kelley [43] searched more than 3 million JPEG images, finding one to two percent of them to be suspicious using steganalysis techniques. It is also reported that Al Qaeda terrorists used pornography as cover media for steganography [43]. With these factors considered, it would be understandable that OSN's would look to avoid steganography being carried out if possible.

Implications for LSB steganography

LSB steganography requires a relatively large cover image in order to create a usable amount of hiding space. Also due to the lossless compression used, bitmaps are often considerably larger than equivalent compressed JPEG images of the same dimensions. It is for this reason that large bitmaps are seen as suspicious on the internet and rarely used [20].

As mentioned, bitmaps can be uploaded to OSN's although they are typically converted to JPEG format and compressed. This conversion destroys individual pixel information, making them unsuitable for steganographic use on OSN's.

Implications for JPEG steganography

The recompression and modifications applied by OSN's causes the DCT coefficients to change, destroying any hidden data. These modifications are typically small enough to not be noticeable by the average user but sufficient enough to remove a hidden message [44].

2.6 Conclusion

This chapter provided an overview of steganography, a technique for hiding data within data, along with its relevant aims and properties. Steganography has proven to be a very useful security tool when used in conjunction with encryption or on its own. Current popular steganographic techniques, LSB Steganography and JSteg and the formats they work with were discussed in order to gain a better understanding as to why they are ultimately unsuitable for use on OSN's.

Clearly traditional steganography algorithms are not going to be suitable for our purposes as they have been designed to favour imperceptibility and capacity over robustness. The following chapter will look at watermarking, a related technique with different objectives.

Chapter 3 – Digital Watermarking

3.1 Introduction

Because watermarking offers the required robustness for our solution, it is discussed in detail in this chapter. Background information on watermarking is given and it is compared to steganography. The Tagging watermarking scheme which serves as a basis for the proposed solution is discussed in detail.

This chapter gives an overview of watermarking and compares it to steganography in section 3.2. section 3.3, looks at the various types of watermarking schemes before section 3.4 focuses specifically on the robust watermarking scheme, Tagging. Section 3.5 looks at how watermarking can be applied to OSN's. The chapter is concluded in section 3.6.

3.2 Overview of Watermarking

Traditional watermarking has been around for centuries; initially used by paper manufacturers to distinguish their papers from one another and more recently it has been adapted to currency too, serving as a counterfeiting prevention mechanism [45]. Digital watermarking has only gained prominence in the last two decades as a tool to protect intellectual property [46]. The watermark is digital data that is robustly, imperceptibly and irremovably embedded into the host data and typically contains information about origin, status, and/or destination of the data [47]. Although it will not prevent the distribution of the content itself, it will enable the content producer to pursue legal actions against the violators of the copyrights if needed [20].

Steganography is similar to watermarking in that both embed a signal within data; they however differ significantly in their application [7]. The differences between the two, as adapted from Fridrich [7], Morkel [14] and Averkiou [45], are summarised below:

	Steganography	Watermarking
Relationship to carrier object	No direct relationship, just a cover for the message	Carrier object is what the watermark protects
Purpose	Hide data in digital media without raising suspicion	Protect intellectual rights of the content producer
Robustness	Desirable, not crucial	Crucial, watermark should be difficult to remove
Capacity	Large capacity is vital, large message potentially need to be stored and transmitted	A large capacity is desirable but not vital, typically just a short signature or something similar is stored
Perceptual Imperceptibility	Very important as communication needs to be secret	Not as important, in fact often advertised in order to deter illegal distribution

Table 3.1: Summary of the differences between Steganography and Watermarking

3.3 Types of Watermarking Schemes

While the main use for robust watermarking is copyright protection, other interesting applications of watermarking have been developed:

- *Fragile Watermarking*: This is a technique developed for image authentication. When an image has been modified, even subtly, the watermark is destroyed – revealing to the content producer that the image has been tampered with. It can also reveal where an image has been tampered with without having to store anything additional along with the image [48].
- *Semi-Fragile Watermarking*: This requires the watermark to survive and remain detectable and authenticable through all image manipulations that in themselves do not damage the image beyond usability [Aggarwal]. It was developed from the recognition that not all modifications to images are malicious in nature. Resizing an image or changing its format for example may not be considered misuse of an image [49].

Because robustness is a vital aspect for our solution, semi-fragile and fragile watermarking schemes would not be suitable and are thus not discussed further.

3.4 Robust Watermarking: Tagging

One of the earliest robust watermarking schemes, Tagging, was suggested by Caronni [8] out of a need to protect and authenticate digital images. For example, a distributor of confidential images would like to prove who is responsible for illegally sharing their images. To achieve this, the distributor would first tag each copy uniquely before distribution. Tagging modifies the image slightly by adding small geometric patterns at brightness levels that are imperceptible [50]. Each instance of an image gets a different arrangement of tags, marking them uniquely. If the distributor comes across an illegal image, they can compare it to their record of tags and identify the original receiver with high probability, even if the image has been modified in some way [51].

Method

The algorithm involves choosing a series of rectangles, known as Tags, in a given image and modulating the brightness of some of these rectangles; with some made brighter and some darker but at levels that are not perceptible to the human eye. The essence of the process can be seen in Figures 3.1 – 3.3 with a fixed tag size of 30x30 for demonstration purposes.



Figure 3.1: Locations are analysed and those that are suitable are chosen for tagging



Figure 3.2: A tagged image with tags exaggerated (20% brightness modulation)



Figure 3.3: A tagged image (2% brightness modulation)

Figure 3.1 demonstrates how suitable blocks are selected for tagging. The image is analysed for tagging locations and depending on the number of expected recipients and the amount of distortion expected, ideal locations are then chosen for embedding. Tag sizes can range from 2×2 to $2n \times 2n$, ($n < \min(X, Y)/2$).

The locations that are avoided are those that are too homogenous (all the pixels in the block are very similar in colour) and those that have sharp breaks in colour, such as an edge (measured using variance). Embedding in these locations would minimise image quality. For more detail, refer to section 4.3.

Figure 3.2 and Figure 3.3 demonstrate the tagging process. Once suitable locations have been identified, a unique tag needs to be embedded. If limited attacks are expected, a random sequence of tags can be used but if enemies are expected to make a number of attacks on the image then Error Correcting Codes (ECC) [52] provide individualised tag sequences with the highest likelihood of recovery.

Tags should:

- Be able to allow differentiation between different recipients.
- Destroy as small amount of information in the original image as possible.
- Be able to be separated easily from the original image to allow for quick and easy verification of the original recipient.
- Not be able to be removed without access to the original image.

Finally tags are added by modulating the brightness of certain blocks. Blocks are made brighter or darker randomly. The modulation should be associated with a recipient and stored in a secure format, to facilitate future comparisons.

Retrieval of Tags

To retrieve the tag data, the brightness of the original image is subtracted from the tagged image; each block's average brightness is calculated and compared to a threshold value to determine if a tag is present and whether it has been made brighter or darker. Once this is done for each block, the owner can compare the tag sequence to a list of stored tags to identify the culprit.

Attacks & Countermeasures

The algorithm takes advantage of the fact that many attacks (modifications made to the image) are not localised and are expected to be spread amongst the entire image. Thus it even performs well against attacks such as changing the image format, scaling brightness/contrast or geometric distortion [51], providing a decent tag recovery in most of these cases.

The following attacks have been considered:

- A strong attack would be enemies *working together to combine their images*. This would involve giving each pixel in the output image the average value between all corresponding pixels, thus flattening out the images profile.
- Solo attackers can *modify image geometry*, such as by shrinking and stretching. These attacks make it difficult to automate detection – the distributor will have to first get the image to the correct dimensions for comparison.
- Solo attackers may *modify image content*, for example by adding noise such as modulating the brightness of parts of the image to obscure tags. They can also compress the image or change its format.

In all the above-mentioned scenarios decent recovery of tags has been demonstrated [8], in most cases well in excess of 66%. Tags of 16x16 pixels and larger are particularly strong, with close to 100% recovery in all tests.

3.5 Applicability of Tagging for OSN's

Having discussed tagging in more detail, we wish to assess its applicability for use on social networks. Tagging certainly presents some desirable properties:

- It is robust enough to withstand a variety of modifications made by OSN's: This is crucial for general applicability to OSN's as each OSN differs in the metadata they embed, the resizing done and the compression performed.
- It can withstand format changes: OSN's typically do not preserve image format, Tagging is robust enough to withstand this.
- It is visually imperceptible: Modulation of block brightness at one or two percent provides good tag retrieval and provides imperceptibility similar to that of LSB in bitmap images [30].

There are, however, a couple of potential drawbacks:

- Tagging has a low capacity: It is designed to uniquely identify image copies for only a handful of recipients, thus does not have a need for a particularly large storage capacity [51].
- It visually degrades the image: Tags are inserted into the image itself by modifying the brightness of blocks, thus changes are made in the spatial domain, reducing image quality [33].

3.6 Conclusion

This chapter provided an overview of watermarking, a technique for preserving intellectual rights in digital media. Watermarking was compared with, and contrasted to steganography. The robust watermarking scheme Tagging, which forms the basis for a solution, was discussed in detail.

Clearly the development of a solution requires several design trade-offs. Along with visual imperceptibility, ensuring robustness against modifications made by OSN's is clearly important too. The solution needs to withstand format conversions, compression and metadata imbedding but to do so, redundancy is required and this means sacrificing capacity to some extent. The next chapter will propose a solution in an attempt to balance these design decisions.

Chapter 4 – The Tagging Steganography Scheme

4.1 Introduction

This chapter discusses how the Tagging can be applied to image steganography on OSN's to successfully hide and retrieve messages. The requirements for a solution are discussed in more detail, focussing on the magic triangle [1] and how to best balance its three components for our purposes. Following that, the technical details of a proposed solution are presented as implemented and tested by the author.

This chapter discusses the requirements for successful steganography on OSN's in section 4.2. Section 4.3, discusses all the technical details of a realisation of these requirements through an implementation as carried out by the author. Section 4.4 concludes the chapter.

4.2 Requirements for successful steganography on OSN's

The aim of steganography on OSN's is to conceal communication by hiding messages in innocuous-looking images in such a way that only the sender and intended recipient knows of the existence of the message [53].

In order to achieve this we would like a solution that:

- Is robust to the modifications made to images by social networks.
- Is undetectable, both statistically and visually.
- Has a large capacity (steganographic bandwidth) for storage of hidden messages.

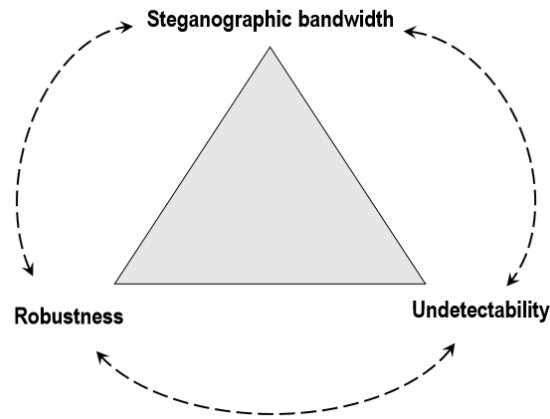


Figure 4.1: The design trade-offs of a steganographic system

Ideally we would like a solution that is robust and hard to detect, but also offers a large capacity. This idea is described by Fridrich [1] as the magic triangle and can be visualised in Figure 4.1. Unfortunately, compromises are required as one property comes at the expense of another. For example, the LSB algorithm as described in section 2.2 has a high capacity but is weak to visual attacks. LSB is also not robust – by modifying a pixel, you are modifying the message. On the other hand, watermarking systems are typically designed to be robust as possible but do not require a large capacity. Also, users may actually know about the existence of the watermark, so undetectability is not a major concern. Figure 4.2 demonstrates this in terms of the magic triangle.

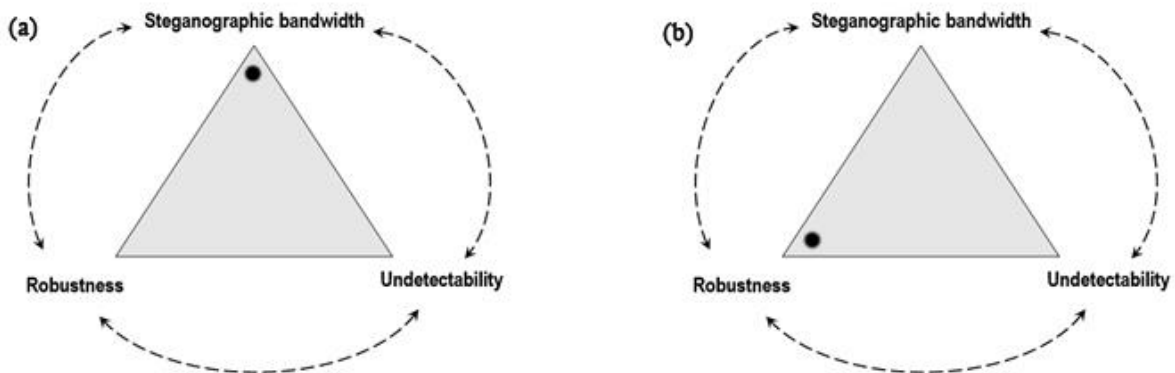


Figure 4.2: The magic triangle for (a) LSB Steganography and (b) watermarking

This dissertation attempts to implement a steganography scheme in such a way as to balance the characteristics described by the magic triangle. The technical details concerning the Tagging Steganography Scheme implementation are discussed in the next section.

4.3 Technical details of the implementation of the Tagging Steganography Scheme

This section describes the technical decisions and considerations that were made in the development of and implementation of the Tagging Steganography Scheme. These considerations include the choice of OSN, the tagging process and the message extraction process.

4.3.1 Choosing an OSN

With more than 350 million photo's posted to Facebook daily [54], it is the most popular OSN and photo sharing website in the world. The decision was thus made to test the characteristics of the Tagging Steganography Scheme on Facebook. The method described would be applicable for other OSN's too; parameters may just have to be adjusted depending on the amount of compression used by these websites.

4.3.2 Ambiguous Terminology

Due to similar terminology being used by Facebook and Tagging, it is necessary to first clarify some terminology in order to avoid ambiguity:

Tagging: For the purposes of this dissertation, Tagging refers to the addition of small geometric patterns to images at brightness levels that are imperceptible in order to conceal hidden messages [50]. This should not be confused with the Facebook function of the same name that enables users to link posts, photos, links and status updates to others within Facebook [54].

Cover Image: For the purpose of this dissertation, a Cover Image is an image that carries a hidden message [55]. It should not to be confused with Facebook's notion of a Cover Image which is a large image at the top of a user's profile [56].

4.3.3 Tagging Considerations for OSN's

The author proposes using Tagging, as introduced by Caronni [8] and described in Chapter 3 in order to embed data in a cover image. Due to the robustness that this method offers, a receiver will be able to reliably retrieve embedded messages from stego-images.

Tagging involves selecting rectangles within the cover image and then modulating the brightness of these rectangles at levels that are imperceptible to the human eye. The original Tagging watermarking scheme has the advantage of a great deal of robustness but suffers from a low capacity. This is largely due to the fact that it needs to withstand a variety of attacks – both known and unknown, and thus a large amount of redundancy (large tags) gives the embedded message the greatest chance of survival [51]. When applying this method to OSN's however, we can increase its embedding capacity significantly because we know the types of attacks that social networks will perform on a cover image and can thus select a tag size and brightness modulation accordingly. These attacks include converting the image to JPEG, resizing the image, compressing the image and embedding metadata [57]. A solution can therefore be designed to withstand these known attacks. Originally Tagging was concerned with these attacks but also attacks such as brightness modulation, cropping or filtering to deliberately remove the watermark. Fortunately these attacks are not currently performed by OSN's so we can 'weaken' the robustness component of our solution slightly in favour of a higher embedding capacity.

4.3.4 Embedding of Tags

Message conversion to binary

In the ASCII character set, each character can be represented by one byte, a number between 0 and 127 [58]. As can be seen in Figure 4.3, the letter 'A' can be represented as 65, the letter 'B' as 66 and so forth. As will be described further, individual tags are made darker or lighter based on whether a 1 or 0 is to be encoded. The message must thus first be converted to the binary equivalent.

"BAD"	Char	Decimal	Binary
	A	65	01000001
	B	66	01000010
	C	67	01000011
	D	68	01000100

== 01000010 01000001 01000100

Figure 4.3: A text string is converted to its byte equivalent representation

Cover Selection

The sender needs to select a cover image which they will use to embed the data. Care is needed to choose an image that is both innocuous (does not stand out as suspicious), and large enough to hold their message.

The original image also needs to be accessible to both sender and receiver as it is required to decode the message. This may seem like a tedious requirement but it is in fact quite trivial due to reverse image searching tools that enable the user to upload a tagged image and find where it originally came from along with other versions of it [59]. Tools such as TinEye and Google offer the ability to perform these searches.

Addition of encryption

Optionally, a message can be encrypted in order to provide further protection [Wang & Wang, Cyber Warfare]. A shared key/password needs to be known beforehand by both parties and can be applied to the message before embedding.

Image Analysis

Suitable locations for tags need to be identified in the chosen cover image. Care needs to be taken to avoid locations that are very flat in colour or those that contain sharp breaks or edges because these are locations that are the most easily detectable [51].

The sender defines a block size for tagging and then divides the image into blocks. As shown in section 5.2, a block size of 8x8 pixels is optimal embedding on Facebook, but this may have to be tweaked for other OSN's depending on the level of compression they apply.

Suitable locations are determined using variance [60], which measures how the perceived brightness of various pixels within a block differ from each another. The formula for determining the perceived brightness of a colour, called HSP, was suggested by [61] and looks as follows:

$$perceived_brightness = \sqrt{(0.299 R^2 + 0.587 G^2 + 0.114 B^2)}$$

Where, R , G and B are the Red, Green and Blue colour components of a pixel.

Variance is then calculated on the *perceived_brightness* of each block of pixels as follows:

$$variance (s^2) = \frac{\sum (x_i - \bar{x})^2}{n - 1}$$

Where:

x_i = *perceived_brightness of one pixel in the block*

\bar{x} = *mean (average) of the whole blocks brightness*

n = *number of pixels in a block*

Blocks with a very high variance are typically those with sharp breaks in colour and are to be avoided. Similarly, blocks with low variance, such as those that are too flat in colour are also avoided. Once the variance for all blocks has been calculated, a choice is made on which of these blocks are actually suitable for embedding as there typically there should be more blocks available than message bits. Which blocks are chosen can be decided through either of two proposed methods: *threshold* or *best-x*.

Threshold

All blocks with a variance between an upper and lower threshold are considered suitable for embedding. These threshold values are discussed in more detail in Chapter 5. The first blocks that meet the criteria are chosen, as demonstrated by Figure 4.3. The disadvantage of this method is that a message may not be able to fit if not enough blocks are between the threshold values, therefore requiring the selection of an alternative cover image or an adjustment of threshold values.

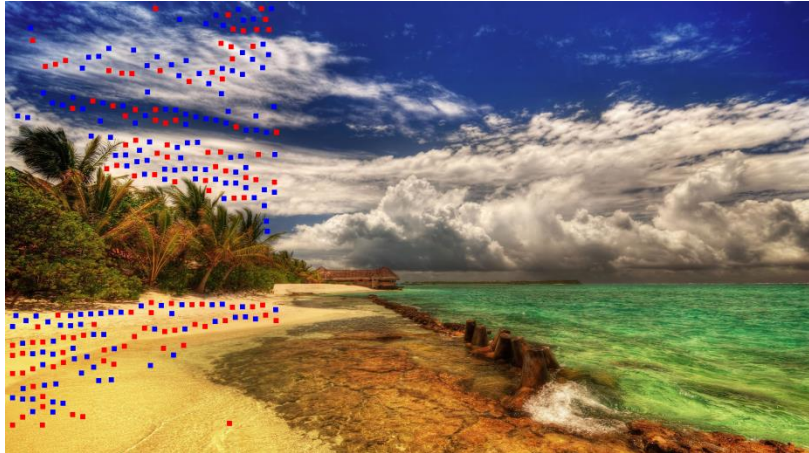


Figure 4.4: Tagging locations are identified using thresholds

Best- x

The x most ideal blocks are chosen, where x is the length of the message. This method has the advantage of being able to embed and disperse the whole message across the image. The disadvantage however is that sub-ideal blocks may be chosen eventually if a too small cover image is chosen or if the message is too large for a single image. The concept can be seen in Figure 4.4.



Figure 4.5: Tagging locations are identified using best- x

Because visual imperceptibility is a vital component of the Tagging Steganography Scheme, the threshold method is preferred.

Integrating Tags

Once locations for tags have been identified, tags can be added by modulating the brightness of the chosen blocks. If a message bit is 1, the block is made darker by decreasing the brightness; if the message bit is 0, the block is made lighter by increasing the brightness.

Care needs to be taken to avoid blocks that will not permit good brightness modulation results. If a block is largely made up of very dark colours (those with RGB values close to (0,0,0)), embedding a 1 (making the block darker), will not help, because the receiver will not because these values cannot be decreased further and the receiver will not be able to detect that a message bit has been embedded. Thus an additional threshold for very light and very dark blocks needs to be defined before message bits are embedded so that those blocks are avoided.

The process for embedding is as follows:

Input: message_bits, shared_img, stego_key (optional)

Output: stego_img

if stego_key

 encrypt message_bits with stego_key

end if

while bits remaining to embed

 get next suitable block

if brightness of block < upper_threshold & brightness of block > lower_threshold

 get message_bit

if message_bit = 1

 modulate block darker

else

 modulate block lighter

endif

endif

end while

Uploading the image

A user can then upload the stego-image to Facebook, or an OSN of their choice. By modifying their privacy settings, they can ensure their posts are visible for the public and not just to their connections [38]. This would imply that the sender and receiver need not even have a social link on Facebook in order to communicate.

4.3.5 Message Extraction

The receiver needs to have knowledge of the senders profile in order to identify it and download the stego-image. Once downloaded, the receiver uploads the tagged image (T) to a reverse-image search in order to identify the source image (S), if they do not already have access to it. The stego-key (K) can be applied if needed. The inverse of the tagging function (F^{-1}) analyses the tags by comparing the block brightness to that of the original image and the message (M) is extracted. The entire process is depicted in Figure 4.6.

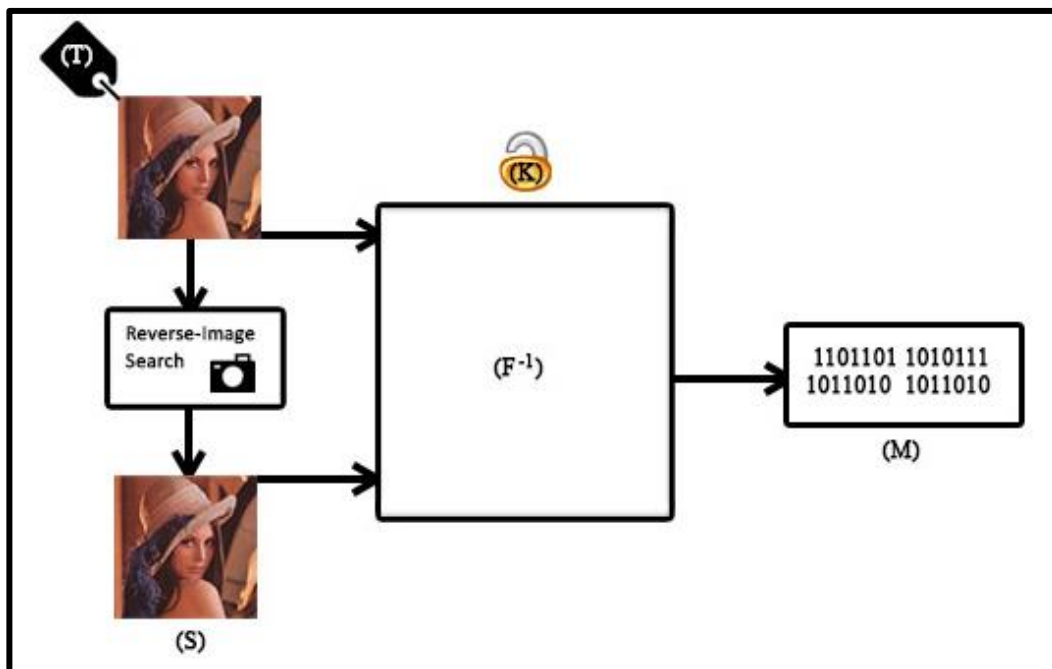


Figure 4.6: *Extraction Process for Tagging on OSN's*

4.4 Conclusion

This chapter described the details concerning the implementation of the Tagging Steganography Scheme. The proposed system has been designed to withstand the modifications Facebook applies to images by providing robustness. The following chapter will discuss experimental results of the system along with its positive and negative aspects.

Chapter 5 – Experimental Results

5.1 Introduction

This chapter describes the experiments that were done using the Tagging Steganography Scheme in detail. The results of the experiments are analysed and a number of parameters are identified based on the experiments that all play a role in creating the balance between imperceptibility, robustness and capacity. These parameters include the block size, percentage of brightness modulation and threshold.

The layout for the rest of this chapter is as follows: Section 5.2 presents experimental results for the Tagging Steganography Scheme; section 5.3 discusses the parameters that make it effective; section 5.4 compares the Tagging Steganography Scheme to popular steganography algorithms, looking at the strong and weak aspects of it. Section 5.5 concludes this chapter.

5.2 Experimental Results

Tests were performed on Facebook using a Java implementation of the Tagging Steganography Scheme. The following message was embedded inside an image:

“Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.”

Details:

- Characters (including spaces): 231
- Bits (tags) required: 1848
- Resolution: 1920 x 1080

- Original Size: 440kb
- Compressed Size (after download from Facebook): 290kb

Parameters:

- Block Size: 8x8 pixels
- Brightness Modulation: 2%
- Threshold: 500 – 1500

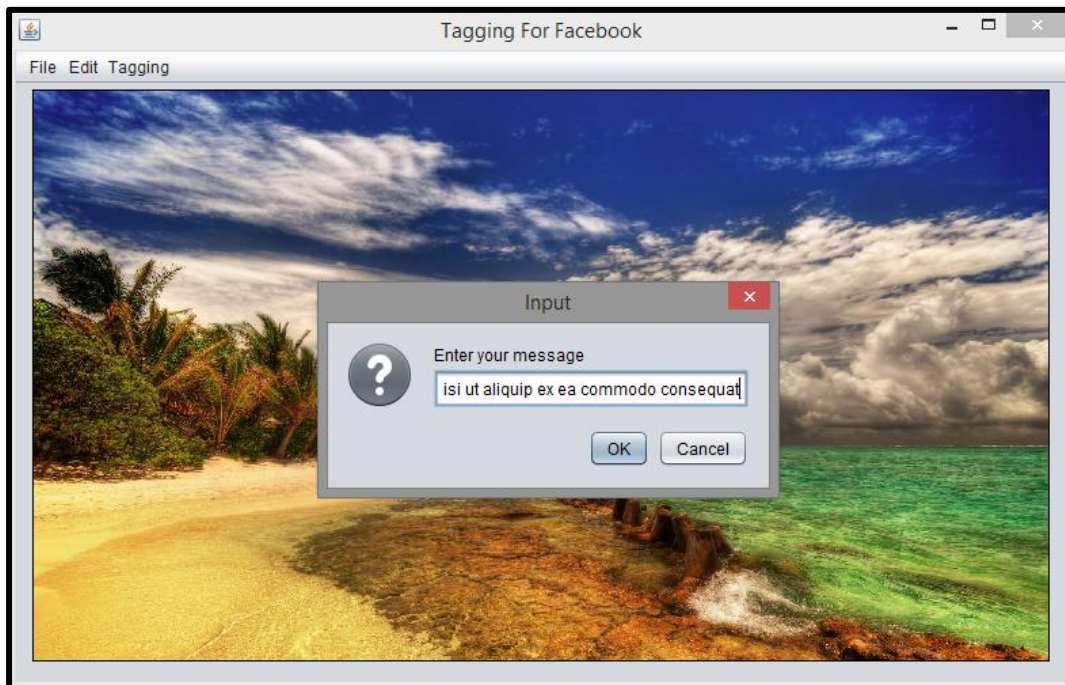


Figure 5.1: A message is embedded using the Tagging Steganography Scheme

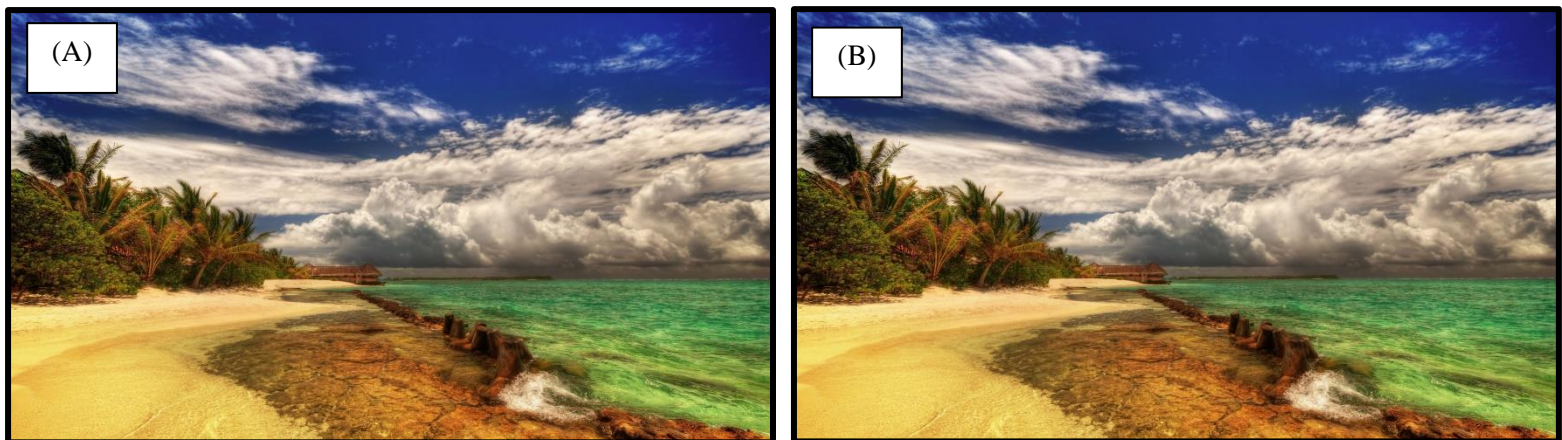


Figure 5.2: The image before tagging (A), and after tagging (B)



Figure 5.3: The stego-image is posted to Facebook and available for download

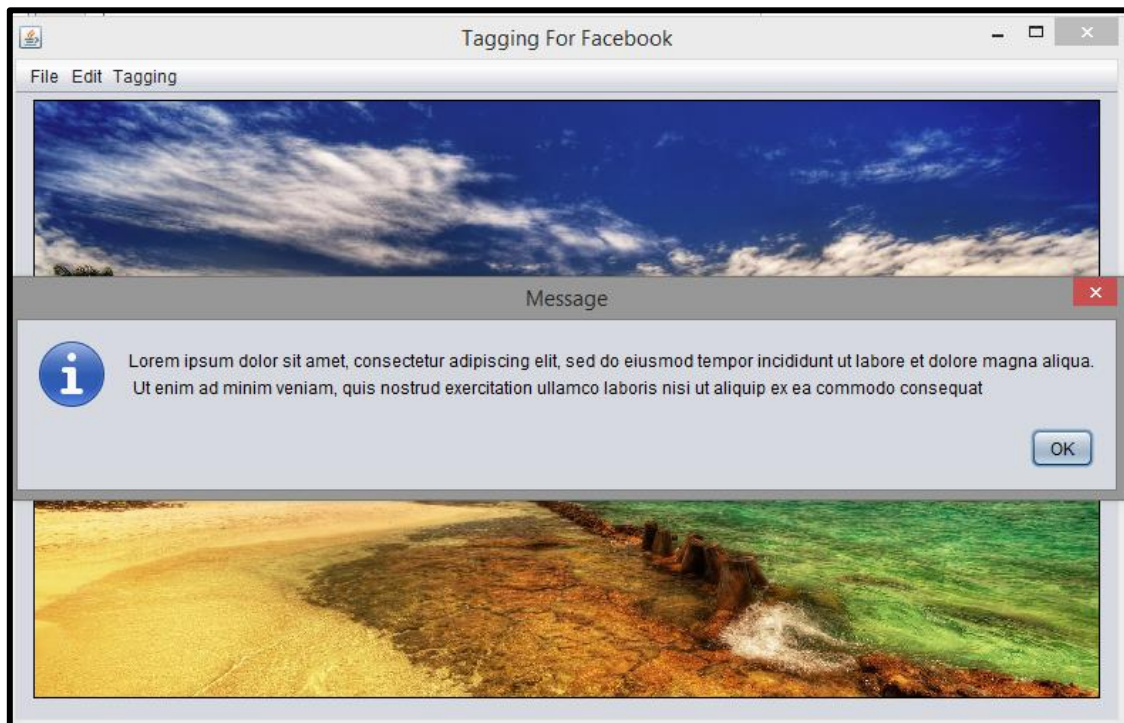


Figure 5.4: The image is analysed and the message is successfully retrieved

The above Figures (5.1 – 5.4) demonstrate how steganography was successfully performed using the proposed scheme. Despite Facebook compressing the image to 65% of its original size (290kb compared with 440kb), 100% of message bits were recovered successfully.

5.3 Parameters

Experiments were done with the Tagging Steganography Scheme implementation to determine the most efficient choice in parameter values that influence the quality of the image and the robustness of the algorithm. These parameters are:

- The percentage each block should be modulated, to ensure that each tag can be retrieved, while attempting to keep tags imperceptible to the human eye.
- The optimal block size for Facebook, in an attempt to ensure 100% retrieval of message bits, while keeping the tags as small as possible.
- The threshold values that are used to determine which blocks are suitable for tagging.

5.3.1 Tag Brightness Modulation

It is vital that tags should be integrated in a way that is visually imperceptible. In a steganographic system, a successful attack is simply to detect hidden communication [7], so it is of great importance that the human eye cannot distinguish the tags. It is for this reason that the parameter that has to take the highest priority is the brightness modulation.

Experiments were done using the Tagging Steganography Scheme implementation and different brightness modulations to determine the ideal percentage with which the brightness should be adjusted to remain imperceptible to the human eye. If the brightness is adjusted too much then it could create visible artefacts in the image, but if the brightness is adjusted too little then the algorithm will not be able to detect the tags. Figure 5.5 shows the results of different brightness modulations applied to part of an image.

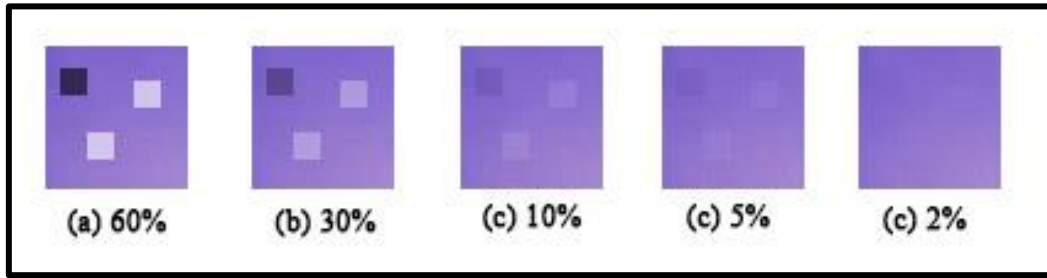


Figure 5.5: Brightness modulation of tags

As can be seen in Figure 5.5, even at 5% brightness modulation, tags are still vaguely visible to the careful observer. It is thus the author's recommendation, that brightness modulation of 2% or less be applied when tagging images for social networks.

5.3.2 Block Size

Another important parameter that is directly related to the capacity of the algorithm is the block size. The larger the block size, the smaller the message length can be. Since Tagging was originally developed to distinguish copies of images uniquely, it is not normally concerned with embedding a large amount of information and tags could therefore be quite large [8]. For use on OSN's however, we require a large capacity in order to embed an entire message on the one hand, but on the other hand the tags should also not be destroyed by OSN modifications.

The size of the tag is dependent on the degree to which the cover image will be compressed. As can be seen in Figure 5.6, JPEG compression disperses a group of colours in order to save on space. In Figure 5.6 (a) the tag is limited to a 3x3 block of pixels, but once compression is applied, pixel values disperse, forming one single texture represented by DCT coefficients rather than RGB values [25]. Figure 5.6 (b) represents typical JPEG compression, while Figure 5.6 (c) demonstrates how an OSN would compress the image further. The images below have been tagged with a 3x3 tag at brightness modulation of 30%. As seen from image (c) from Figure 5.6, the tag survived the OSN compression and is still recoverable.

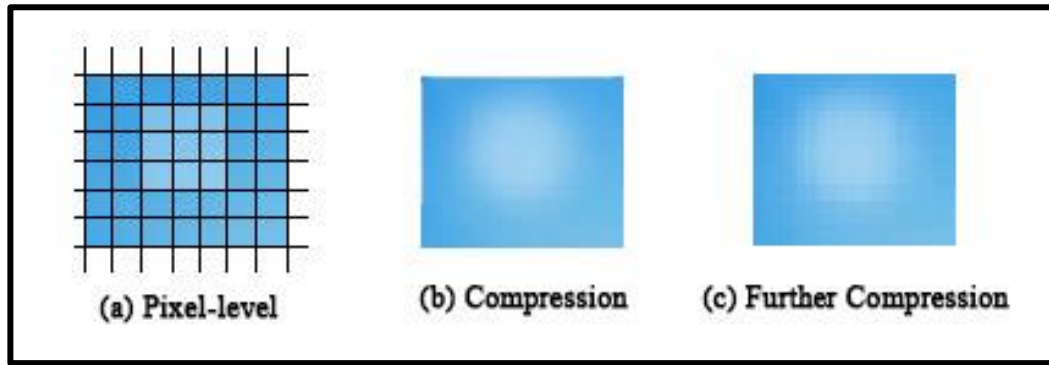


Figure 5.6: JPEG Compression degrades image information

However, when using a brightness modulation level of 2% or less, it was found that recovering 100% of tags is not easy due to the dispersion being done on very similar colours. For the original application of Tagging as a watermarking scheme, 100% recovery of tags was not necessary, as tags were often embedded redundantly. Only enough tags needed to be retrieved to prove the guilt of one party over another. For use on OSN's however, 100% recovery of tags is vital as each tag represents a message bit. Changing a single bit can change the entire message.

The following table represents the tests carried out on block size by the author in order to determine which would be most suitable for use on OSN's. The tests were performed embedding 128 characters (1024 bits) at a brightness modulation of 2% in 100 different HD images.

Block Size	Average Bits Recovered	Average Recovery Percentage
2x2	747	73%
4x4	965	94%
6x6	1005	98%
8x8	1024	100%
12x12	1024	100%

Table 5.1: Recovery rate of various block sizes

The results from Table 5.1 show that the average recovery percentage increases with an increase in block size. Thus with an increased redundancy comes an increased chance of successful tag recovery. There is thus a trade-off between the size of the tags, in other words the capacity

provided by the algorithm, and the amount of tags that are recoverable. Because 8x8 pixels is the smallest block size with 100% recovery, it is recommended for use on Facebook.

5.3.3 Thresholds

Threshold values are used to determine which blocks are suitable for embedding and which are not. As can be seen by Figure 5.7, locations with sharp breaks and those with a lot of change in brightness/colour should be avoided because tags placed in these areas will be more noticeable [51].

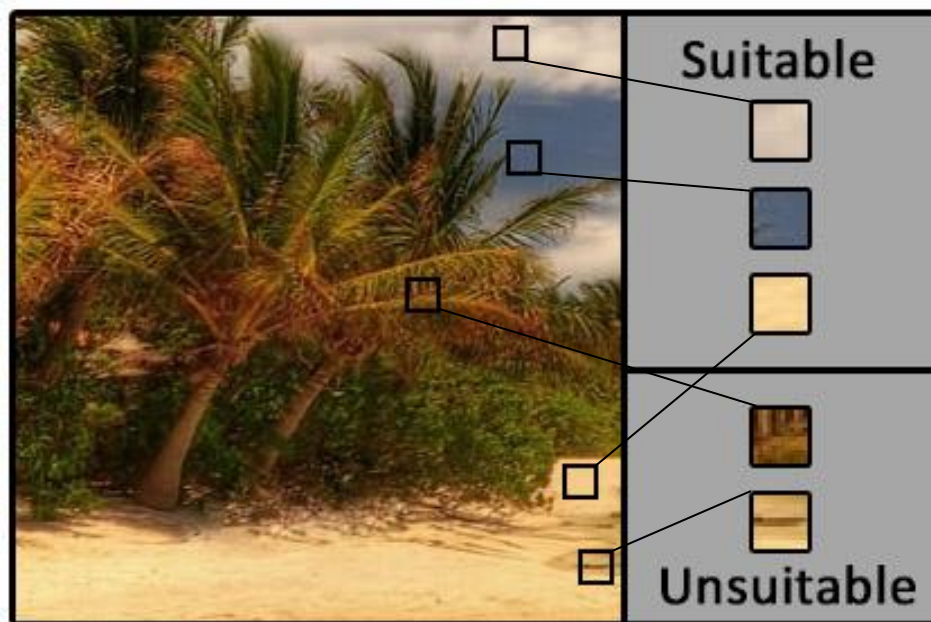


Figure 5.7: Suitable and unsuitable tagging locations as determined by thresholds

The variance in brightness within a block is visualised in Table 5.2. The histograms for the unsuitable blocks are very spread out, indicating a large difference in brightness. This is also reflected in their considerably higher variance (See Section 4.3).




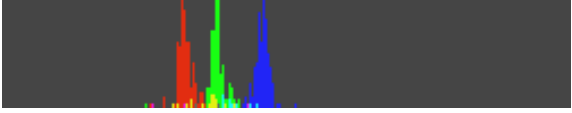

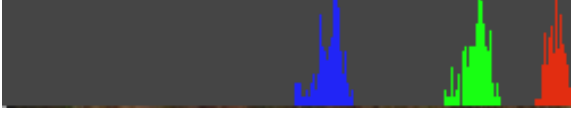

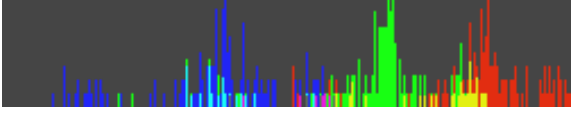

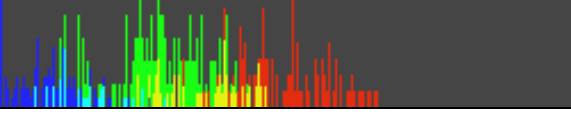
Block	Histogram	Variance	Suitable
		251.86	Yes
		201.64	Yes
		804.29	Yes
		3171.94	No
		1949.22	No

Table 5.2: Histograms and variance of different blocks

It is difficult to define exactly what variance thresholds to be used as one does not want to limit locations to the extent that very few message bits can be embedded, but one does also not want embedding performed in unsuitable locations since this would result in visible artefacts. It is recommended that a variance of between 150 (lower threshold) and 1500 (upper threshold) be used, as this provides decent capacity while avoiding unsuitable areas to a large extent. Further research could be done to define these thresholds with more authority.

5.4 Analysis of the Tagging Steganography Scheme

In research, a few authors have suggested the need for a system that can successfully perform steganography on OSN's [44; 57; 62]. Castiglione [57] has attempted to solve the same problem by using image filenames as cover objects to circumvent image manipulation issues. This method however has a very low capacity and would require a large number of images in order to be effective.

The implementation of Tagging thus provides a solution to using steganography on OSN's, while maintaining imperceptibility, robustness and capacity. The Tagging Steganography Scheme can reliably embed and retrieve data from images despite modifications on social networks without visual detection. One area where the Tagging Steganography Scheme is not as effective as traditional steganography algorithms is capacity, which has had to be sacrificed in favour of robustness against OSN modifications. A possible drawback of the Tagging Steganography Scheme is that an image has to first be analysed for locations prior to embedding so that the user knows whether the image will have the capacity to accommodate the message or not. This is similar to JPEG steganography where there is uncertainty how many suitable coefficients will be available for embedding. Another possible inconvenience of the Tagging Steganography Scheme when compared with traditional steganography is the need to compare the downloaded image to the original image in order to locate the tags. This however can be solved by using reverse-image searching.

Despite these shortcomings, the proposed solution is a useful tool for performing steganography in environments where robustness is required, such as those that perform image conversion, compression and resizing. Due to its limited capacity, it is suggested that for less-hostile environments, traditional steganography algorithms would be more useful.

5.5 Conclusion

This chapter looked at the important parameters relevant to the proposed solution, with brightness modulation; block size and threshold values discussed in more detail. Recommendations for these parameters were provided based on the author's experiments on Facebook. The proposed solutions strengths and weaknesses were described and it was compared to other current steganography algorithms, in terms of its purpose and characteristics.

Chapter 6 – Conclusion

6.1 Summary

Steganographers are always looking for new mediums and platforms on which they can perform steganography. Online Social Networks (OSN's) are one such platform that provides large-scale public sharing of digital media over the internet; certainly providing an environment that would be appealing for steganography. Unfortunately these websites modify images uploaded to them in a number of ways, making current image steganography techniques difficult to use on them.

This dissertation aimed to address this by adapting the robust watermarking scheme *Tagging* for steganographic use on OSN's. The Tagging Steganography Scheme aimed to provide suitable robustness to survive OSN modifications, while maintaining visual imperceptibility and adequate storage capacity. After testing the solution extensively on Facebook, it is safe to say that the solution met these objectives. While the solution cannot compare to traditional steganography algorithms in terms of capacity, it provides considerable robustness against a variety of attacks which makes it useful for use on OSN's or similarly hostile environments.

6.2 Future Research

Some ideas for future research include:

Steganalysis for Tagging

This dissertation proved the Tagging Steganography Scheme could perform steganography without producing visual artefacts. However, tests for statistical detectability were not pursued. The Chi-Squared test [63] and visual attacks [13] amongst others could possibly be used.

Threshold calculation

This dissertation suggested thresholds for the selection of tags based on what appeared to give a good balance of capacity and undetectability. In future work, these thresholds could be investigated and defined with more authority.

References

- [1] Fridrich, J. 1998. "Applications of Data Hiding In Digital Images", Proceedings of ISPACS'98 Conference, Melbourne, Australia.
- [2] Kahn, D. 1996. "The history of steganography", Springer, Lecture Notes in Computer Science Volume 1174.
- [3] Morkel, T., Eloff, J.H.P. & Olivier, M.S. 2005. "An Overview of image steganography", ISSA.
- [4] Oxford English Dictionary. 1989. Oxford: Oxford University Press.
- [5] Nielsen Online Report. 2009. "Social networks & blogs now 4th most popular online activity", [Online] Available From: http://www.nielsen.com/us/en/press-room/2009/social_networks_.html (Accessed: 18 October 2014)
- [6] Stegano.net. 2014. "Network Steganography Principles", [Online] Available From: <http://stegano.net/tutorial/net-steg.html> (Accessed: 18 October 2014)
- [7] Fridrich, J. 2009. "Steganography in digital media: principles, algorithms, and applications", Cambridge, U.K.: Cambridge Univ. Press.
- [8] Caronni, G. 1995. "Assuring ownership rights for digital images", Springer Verlässliche IT-Systeme.
- [9] Norman, B. 1973. "Secret Warfare", Acropolis Books, Washington, D.C.
- [10] Zim, H.S. 1948. "Codes and Secret Writing", William Morrow, New York.

- [11] Simmons, G.J. 1983. "The Prisoners' Problem and the Subliminal Channel", Advances in Cryptology, Springer.
- [12] Tseng, Y.C., Chen, Y.Y., Pan, H.K. 2002. "A secure data hiding scheme for binary images", IEEE Transactions on Communications, Volume 50, Issue 8.
- [13] Das, S., Das, S., Bandyopadhyav, B. 2011. "Steganography and Steganalysis: different approaches", International Journal of Computers, Information Technology and Engineering (IJCITAE), Vol. 2, No 1.
- [14] T. Morkel. 2012. "Image steganography applications for secure communication", University of Pretoria.
- [15] Dunbar, B. 2002. "A detailed look at Steganographic Techniques and their use in an Open-Systems Environment", SANS Institute.
- [16] Anderson, R.J., Petitcolas, F.A.P. 1998. "On the limits of steganography", IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection 16 No.4
- [17] Artz, D. 2002. "Digital Steganography, Hiding Data within Data", IEEE, Internet Computing (Volume 5, Issue 3).
- [18] Kawaguchi, E., Eason, R.O. 1999. "Principles and applications of BPCS steganography", Multimedia Systems and Applications (Volume 3528), Boston, MA
- [19] Wang, H., Wang, S. 2004. "Cyber warfare: steganography vs. steganalysis", Communications of the ACM, 2004.

- [20] Krenn, J.R. 2004. "Steganography and Steganalysis" [Online] Available From: <http://bandwidthco.com/whitepapers/compforensics/steganography/Steganography%20and%20Steganalysis.pdf> (Accessed: 22 October 2014)
- [21] Hayati, P. 2007. "A Survey of Steganographic and Steganalytic Tools for the Digital Forensic Investigator". Workshop of Information Hiding and Digital Watermarking to be held in Conjunction with IFIPTM, Moncton, New Brunswick, Canada
- [22] Owens, M. 2002. "A discussion of covert channels and steganography", SANS institute 1 (2002): 1-18.
- [23] Howard, P.G. 1993. "Fast and Efficient Lossless Image Compression", Proceedings of the 1993 IEEE Data Compression Conference (DCC '93), Snowbird, UT.
- [24] Sachnev, V., Hyoung Joong, K., & Zhang, R. 2009. "Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding." Proceedings of the 11th ACM workshop on Multimedia and security.
- [25] Wallace, G.K. 1992. "The JPEG still picture compression standard", Consumer Electronics, IEEE Transactions on 38.1: xviii-xxxiv.
- [26] Westerman, L.A. 2001. "Rendering of YCBCR images on an RGS display device", U.S. Patent No. 6,326,977.
- [27] Marshall, D. 2001. "The Discrete Cosine Transform", [Online] Available From: <http://www.cs.cf.ac.uk/Dave/Multimedia/node231.html> (Accessed 01 November 2014).
- [28] Snell & Wilcox. 2002. "MPEG Encoding Basics" [Online] Available From: <http://www.media-matters.net/docs/resources/Digital%20Files/MPEG/MPEG%20Encoding%20Basics.pdf> (Accessed 01 November 2014).

- [29] Pennebaker, W.B., and Mitchell, J.L. 1993. "JPEG: Still image data compression standard". Springer.
- [30] Upham, D. 2000. "Jsteg", [Online] Software available From:
<http://zooid.org/~paul/crypto/jsteg>
- [31] Pujar, J. H., & Kadlaskar, L. M. 2010. "A NEW LOSSLESS METHOD OF IMAGE COMPRESSION AND DECOMPRESSION USING HUFFMAN CODING TECHNIQUES", Journal of Theoretical & Applied Information Technology, 15.
- [32] Fridrich, J., Pevný, T., & Kodovský, J. (2007). "Statistically undetectable jpeg steganography: dead ends challenges, and opportunities", In Proceedings of the 9th workshop on Multimedia & security (pp. 3-14). ACM.
- [33] Kodovsky, J., & Fridrich, J. 2010. "Quantitative structural steganalysis of Jsteg", Information Forensics and Security, IEEE Transactions on, 5(4), 681-693.
- [34] Provos, N., & Honeyman, P. 2003. "Hide and seek: An introduction to steganography" Security & Privacy, IEEE, 1(3), 32-44.
- [35] Westfeld, A., & Pfitzmann, A. 2000. "Attacks on steganographic systems", In Information Hiding (pp. 61-76). Springer Berlin Heidelberg.
- [36] Provos, N. 2001. "Defending Against Statistical Steganalysis". In Usenix Security Symposium (Vol. 10, pp. 323-336).
- [37] Westfeld, A., & Pfitzmann, A. 2001. "High capacity despite better steganalysis (F5—a steganographic algorithm)", In Information Hiding, 4th International Workshop (Vol. 2137, pp. 289-302).

- [38] Facebook. 2014. "Permissions" [Online] Available From: <https://developers.facebook.com/docs/facebook-login/permissions> (Accessed 02 November 2014).
- [39] Bird, S. W., & Veconi Jr, G. J. 1994. "Computer implemented method and system for storing and retrieving textual data and compressed image data" U.S. Patent No. 5,325,297. Washington, DC: U.S. Patent and Trademark Office.
- [40] TeleCrunch. 2014. "Facebook Says Paper Users Browse 80 Stories a Day", [Online] Available From: <http://techcrunch.com/2014/04/18/facebook-paper-users/> (Accessed 02 November 2014).
- [41] FreeDigitalPhotos.net. 2014. "Avoiding Facebook Image Compression", [Online] Available From: <http://www.freedigitalphotos.net/blog/tutorials/avoiding-facebook-image-compression/> (Accessed 02 November 2014).
- [42] Hosmer, C. 2006. "Discovering hidden evidence" Journal of Digital Forensic Practice, 1(1), 47-56.
- [43] Kelley, J. 2001. "Terror groups hide behind Web encryption", USA today, 5.
- [44] Beato, F., De Cristofaro, E., & Rasmussen, K. B. "Undetectable Communication: The Online Social Networks Case".
- [45] Averkiou, M. "Digital Watermarking".
- [46] Podilchuk, C. I., & Delp, E. J. 2001. "Digital watermarking: algorithms and applications", Signal Processing Magazine, IEEE, 18(4), 33-46.
- [47] Hartung, F., & Kutter, M. 1999. "Multimedia watermarking techniques", Proceedings of the IEEE, 87(7), 1079-1107.

- [48] Fridrich, J., Goljan, M., & Baldoza, A. C. 2000. "New fragile authentication watermark for images", In Image Processing. Proceedings. 2000 International Conference on (Vol. 1, pp. 446-449). IEEE.
- [49] Sun, R., Sun, H., & Yao, T. 2002. "A SVD-and quantization based semi-fragile watermarking technique for image authentication". In Signal Processing, 2002 6th International Conference on (Vol. 2, pp. 1592-1595). IEEE.
- [50] Duan, F. Y., & King, I. 1999. "A Short Summary of Digital Watermarking Techniques for Multimedia Data", Department of Computer Science & Engineering, The Chinese University of Hong Kong. Shatin, NT, Hong Kong, China.
- [51] Caronni, G., & Schuba, C. 2001. "Enabling hierarchical and bulk-distribution for watermarked content". In Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual (pp. 277-285). IEEE.
- [52] Peterson, W.W., & Weldon, E. J. 1972. "Error-correcting codes", MIT press.
- [53] Hanzlik, P. 2011. "Steganography in Reed-Solomon Codes", Luleå University of Technology.
- [54] CrossCreativeMarketing.com. 2014. "What is Tagging on Facebook and Why is it Important for Your Small Business", [Online] Available From: <http://crosscreativemarketing.com/what-is-tagging-on-facebook-and-why-is-it-important-for-your-small-business> (Accessed 03 November 2014).
- [55] Hussain, M., & Hussain, M. 2013. "A Survey of Image Steganography Techniques".
- [56] Facebook. 2014. "Facebook Cover Images" [Online] Available From: <https://www.facebook.com/help/388305657884730/> (Accessed 03 November 2014).

- [57] Castiglione, A., D'Alessio, B., & De Santis, A. 2011. "Steganography and secure communication on online social networks and online photo sharing". In Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 International Conference on (pp. 363-368). IEEE.
- [58] Cerf, V. G. 1969. "ASCII format for network interchange".
- [59] CBROnline.com. 2014. "Top 5 Reverse Image Search Engines", [Online] Available From: <http://www.cbronline.com/news/social/top-5-reverse-image-search-engines-4276218> (Accessed 03 November 2014).
- [60] Shapiro, S. S., & Wilk, M. B. 1965. "An analysis of variance test for normality (complete samples)". *Biometrika*, 591-611.
- [61] Finley, D.R. 2006. "HSP Color Model — Alternative to HSV (HSB) and HSL" [Online] Available From: <http://alienryderflex.com/hsp.html> (Accessed 04 November 2014).
- [62] Martini, A., Zaharis, A., & Ilioudis, C. 2009. "Data hiding in the SWF format and spreading through social network services", *Digital Forensics & Incident Analysis*—(WDFIA 09), Athens.
- [63] Greenwood, P. E. 1996. "A guide to chi-squared testing", (Vol. 280). John Wiley & Sons.