Digital Image Steganalysis Schemes for Breaking Steganography

Kanchan Patil
Department of I.T.
SSSIST,Sehore,
Bhopal.

Ravindra Gupta
Department of I.T.
SSSIST,Sehore,
Bhopal.

Gajendra Singh
Department of I.T.
SSSIST,Sehore,
Bhopal.

ABSTRACT
Steganography is the art and science of secret communication, aiming to conceal the existence of a communication. Steganography in the modern day sense of the word usually refers to information or a file that has been concealed inside a digital Image, Audio/Video file. Information Security is becoming an inseparable part of Data Communication through Internet. In order to address this Information Security, Steganography plays an important role. In contrast to steganography, steganalysis is focused on detecting, tracking, extracting, and modifying secret messages transmitted through a covert channel. The digital media steganalysis is divided into three domains, which are image steganalysis, audio steganalysis, and video steganalysis. In this paper we are discussing about the digital image steganalysis technique for breaking steganography algorithm.

Keywords
Steganography, Steganalysis, Cryptography, Cryptanalysis, Ciphertext, Image, Cover Media.

1. INTRODUCTION
The Internet has revolutionized the modern world and the numerous Internet based applications that get introduced these days add to the high levels of comfort and connectivity in every aspects of human life. Internet is used for various purposes – ranging from accessing information for educational needs to financial transactions, procurement of goods and services [1]. As the modern world is gradually becoming "paperless' with huge amount of information stored and exchanged over the Internet, it is imperative to have robust security measurements to safeguard the privacy and security of the underlying data.

Cryptography techniques [2] have been widely used to encrypt the plaintext data, transfer the ciphertext over the Internet and decrypt the ciphertext to extract the plaintext at the receiver side. However, with the ciphertext not really making much sense when interpreted as it is, a hacker or an intruder can easily perceive that the information being sent on the channel has been encrypted and is not the plaintext. This can naturally raise the curiosity level of a malicious hacker or intruder to conduct cryptanalysis attacks on the ciphertext[2].

It would be rather more prudent if we can send the secret information, either in plaintext or ciphertext, by cleverly embedding it as part of a cover media (for example, an image, audio or video carrier file) in such a way that the hidden information cannot be easily perceived to exist for the unintended recipients of the cover media. This idea forms the basis for Steganography, the art of invisible communication. Its purpose is to hide the very presence of communication by embedding messages into innocuous-looking cover objects. In today's digital world, invisible ink and paper have been replaced by much more versatile and practical covers for hiding messages – digital documents, images, video, and audio files. As long as an electronic document contains perceptually irrelevant or redundant information, it can be used as a "cover" for hiding secret messages. In this paper, we deal solely with covers that are digital images stored in the JPEG format. Each steganographic communication system consists of an embedding algorithm and an extraction algorithm. To accommodate a secret message, the original image, also called the cover-image, is slightly modified by the embedding algorithm. As a result, the stego-image is obtained. An extraction algorithm is used to get secret message from stego-images.

Steganography protects the intellectual property rights and enables information transfer in a covert manner such that it does not draw the attention of the unintended recipients. Therefore information hiding has been a hot research issue in recent years. Early research has focused on steganography to establish secret channels between two parties. In today's modern world, this has changed thoroughly, however, and we must take a brand new view about steganography by using steganalysis techniques.

Steganalysis is the science of detecting the presence of hidden data in the cover media files and is emerging in parallel with steganography. The method is secure if the stego-images do not contain any detectable artifacts due to message embedding. In other words, the set of stego-images should have the same statistical properties as the set of cover-images. If there exists an algorithm that can guess whether or not a given image contains a secret message with a success rate better than random guessing, the steganographic system is considered broken. In this paper we will discuss various schemes of steganalysis for breaking steganography algorithms.

2. STEGANOGRAPHIC TECHNIQUES
In the last few years the theoretical foundations of information hiding has advanced very rapidly. Modeling the information hiding process as one of communications security produced improved information hiding algorithms as well as accurate models of the channel capacity and error rates. At the same time, steganography security, i.e. the ability of information hiding to serve in a scenario where the presence of an enemy explicitly aiming at nullifying the hidden information goals, whatever they are, has been recognized as one of the main open issues steganographic techniques face with.

For all the steganographic systems, most vital and elementary requirement is the undetectability. The hidden message should not be detected by any other people. Moreover, the cover message with hidden message i.e. stego-media are Indistinguishable from the original ones i.e. cover-media. The

cover-media and stego-media should appear identical under all possible statistical attacks and the embedding process should not degrade the media fidelity. The difference between stego-media and the cover-media should be imperceptible for visual attacks.

There are two types of algorithms 1) spatial domain, 2) transform domain.

2.1 Spatial Domain

In spatial domain we actually consider the image as a 2-d function of a Cartesian coordinates. Each pixel in the image is represented as in terms of coordinates & processing of the image is carried out on each pixel. Various algorithms of such type are LSB embedding, Pixel value differencing, Tri way pixel value differencing etc.

2.1.1 Least Significant Bit Substitution Techniques (LSB)

It is one of the earliest stego-systems to surface were those referred to as Least Significant Bit Substitution techniques, so called because of how the message data m is embedded within a cover image c. In computer science, the term Least Significant Bit (LSB) refers to the smallest (right-most) bit of a binary sequence.

2.1.2 Hide & Seek

The simplest form of image steganography is the method known as Hide & Seek which replaces the LSBs of pixel values (also referred to as the spatial domain) with the bits from the message bit stream. The algorithm is so straightforward that it does not require a key to be implemented. Whilst this makes things a lot simpler to program and exchange the secret, it does mean that the security lies solely in the algorithm. If a key were used, then it might still be impossible for the adversary to decode the hidden message, as the key would usually index the manipulated regions of the image. In the case of the Hide & Seek algorithm however, the adversary simply needs to understand how the algorithm works, and they will be able to decrypt the message.

2.2 Transform Domain

In transform domain we actually consider the image in terms of frequency components. The part of the image where edges are present is termed as high frequency components having larger variations in pixel intensity values & the smoother areas are termed as low frequency components where the pixel intensity values don't differ much. The algorithm of such type is Discrete cosine transforms.

2.2.1 JStag

The JSteg algorithm was developed by Derek Upham and is esentially a carbon copy of the Hide & Seek algorithm discussed in section 2.1.2, because it employs sequential least significant bit embedding. In fact, the JSteg algorithm only differs from the Hide & Seek algorithm because it embeds the message data within the LSBs of the DCT coefficients of c, rather than its pixel values.

2.2.2 OutGuess

In much the same way that embedding the message data sequentially using the Hide & Seek method was not considered very secure, neither was the fact that the JSteg algorithm embedded in the same fashion. The first version of OutGuess, designed by Neils Provos [18], improved the JSteg algorithm by scattering the embedding locations over the entire image according to a PRNG on image c derived using seed k. This is very similar to the way that the randomized embedding approach improved the Hide & Seek algorithm.

2.2.3 F3

As an alternative to the OutGuess 0.2 algorithm, AndreasWestfeld designed an algorithm called F3 [27] which were considered even more secure. The reason for this is that it did not instantiate the same embedding process as the JSteg and OutGuess algorithms. Instead of avoiding embedding in DCT coefficients equal to 1, the F3 algorithm permitted embedding in these regions, whilst it would still avoid embedding in zeros and the DC coefficients. The algorithm still embedded the message data sequentially within c.

2.2.4 F4

The main pitfall with F3 was the fact that it effectively embedded more zeros than ones as a result of the shrinkage mechanism. This meant that when the statistical properties of the stegogrammes are examined through its histogram for example, some artifacts of embedding became apparent. This is much the same as what happened in the JSteg implementation except a slightly different pattern is derived. In addition to this, steganalysts also found that more odd coefficients existed in F3 stegogrammes than even coefficients. This now meant that there were two deficiencies that could be examined when viewing the histogram of a suspect image. F4 was developed to remove these properties such that the histogram would appear similar to that of a clean image.

2.2.5 F5

The F5 algorithm [7] is predominantly the same as the F4 algorithm, at least in terms of its strategy for encoding the message data. However, the F5 algorithm was designed in an attempt to improve on the F4 algorithm by minimising the disturbance caused on c when embedding the message data. This was achieved by introducing matrix encoding, and the algorithm was the first known stego-system to make use of this technique. We will not review matrix encoding in great detail as it is rarely used for steganography, however we should be aware that it significantly decreases the necessary number of changes required for embedding the message data.

3. STEGNALYSIS FOR BREAKING STEGANOGRAPHY

Section 2 gives an illustration of just a few steganographic algorithms that can be used to embed a secret message within an image. If we are to assume that steganography is used with ill-intent (such as terrorism) then it is imperative that we continually develop steganalytical schemes capable of breaking steganography. Steganalysis is an extremely difficult science, as it relies on insecure steganography. If steganography is to be successful, it should leave no indication that a secret message exists. Thus, if the model has been created successfully, it should be a difficult task for any third party to spot that tampering has occurred. Jessica Fridrich [5] suggests that "the ability to detect secret messages in images is related to the message length". This statement is based on the logic that a small message embedded within a large carrier will result in a small percentage of manipulations, and therefore it will be much harder to spot any artifacts within the stegogramme.

Of course, the success of steganalysis also depends on what information the steganalyst has to work from. There are two main classifications of steganography - targeted, and blind. We will focus on how they can be used to combat the steganographic algorithms discussed in Section 2, as well as introducing blind steganalysis techniques.

3.1 Trageted Steganalysis

Targeted steganalysis works when a method designed for identifying a specific steganographic algorithm has been developed [1]. For example, embedding within pixel values leaves patterns that can be searched for with suspicious files. If the steganalyst is sure that covert communications are taking place, and also knows of a possible method for how a secret message can be embedded, then it should be a fairly trivial task to summaries if the file contains this type of steganography or not. This section presents some basic steganalytical schemes associated with "targeted" steganalysis, including visual, structural, and statistical attacks.

3.1.1 Visual attacks

Visual Attacks are widely regarded as the simplest form of steganalysis. As the name suggests, a visual attack largely involves examining the subject file with the naked eye to identify any obvious inconsistencies. Of course, the first rule of steganography is that any modifications made to a file should not result in quality degradation, so a good steganographic implementation will create stegogrammes that do not look any more suspicious than the cover Work - at least not at face value. However, when we remove the parts of the image that were not altered as a result of embedding a message, and instead concentrate on the likely areas of embedding in isolation, it is usually possible to observe signs of manipulation. It can therefore be argued that the key aspect of a successful visual attack is to correctly determine which features of the image can be ignored (redundant data), and which features should be considered (test data) in order to to test the hypothesis that a suspect image contains steganography. An incorrect choice can lead to an increase in false-negatives, which is something a steganalyst would want to avoid. As a result, it is highly likely that all permutations of possible redundant and test data sets will be analysed such that the steganalyst is in the strongest position to make an informed conclusion. Visual attacks on sequential and randomized Hide & seek alogorithms are the examples.

Visual attacks are very time- consuming to produce test images for several possible methods of embedding, and that is before they are perceptually analysed. If a steganalyst wishes to exhaust every type of embedding strategy, they would need to look at thousands of images to consider the likelihood that a single suspect image is a stegogramme. This is obviously an inefficient methodology, and is often the reason why other steganalytical methods are preferred.

3.1.2 Structural attacks

Structural attacks are designed to take advantage of the high-level properties that are known to exist for a particular steganographic algorithm. For example, version 4.1 of Hide & Seek was forced to operate only on images that were of size 320 x 480 pixels [19]. Similarly, StegoDos operated only on images of size 320 x 200 pixels [10]. This means that a steganalyst that happens to intercept images of either of these sizes, can immediately flag them as suspicious.

Structural attacks rarely analyse each image on its own merits. Instead, the images are scanned to see if they contain any of the known side-effects for various steganographic algorithms. Images that contain these properties are often subjected to further investigation. There are sometimes cases where the image may possess symptoms of steganography when it is actually perfectly innocent. For example, computer generated images are likely to have a different colour composition than those of natural life because they are not influenced by the same elements such as light, shadowing, and sampling. Computer generated images may therefore appear structurally similar to what is expected for a stegogramme, but they do not necessarily contain hidden messages; this is why a more thorough investigation usually follows a structural attack. Structural attacks on file size and palette-based steganography are the examples.

Structural attacks are arguably more important to steganalysts than visual attacks because they can be applied against a wider range of embedding techniques. The attacks work best when the steganalyst has access to a known stegogramme. In these instances, the steganalyst can probe the image for inconsistencies, thus producing a feature set that is known to be assoiciated with stegogrammes. A structural attack can then be instantiated by inspecting suspicious images for these features; those that contain the same properties are likely to be stegogrammes. With this in mind, structural attacks are rarely used as a means of proving that an image contains steganography, rather they highlight images that contain signs of embedding.

3.1.3 Statistical attacks

In mathematics, the study of statistics makes it possible to determine whether some phenomenon occurs at random within a data set [24]. Usually, a theory would be constructed that seemingly explains why the phenomenon occurs, and statistical methods can then be used to prove this theory to be either true or false. If we think about the data structure for a stegogramme, we can begin to see how statistics can be useful for steganalysis when proving whether or not the image contains a hidden message. A stegogramme can be broken down into two data sets: image data, and message data. The image data relates to the information regarding the physical image that we can see, and will typically relate to pixel values that point towards the colours used in that region of the image. The message data on the other hand, relates to information regarding the secret message, and - if encrypted - it is typically more randomly composed than image data. It can safely be derived that the message data is more random than image data, and this is where statistical attacks usually operate. Whilst there is usually far less message data than image data, the small percentage of randomness created by the message data is enough to invoke an attack. There are several methods that are known to prove the existence of a hidden message via statistical approaches; each aimed at identifying signs of embedding for specific stegosystems. One of them is chi-squared test. The test makes it possible to compare the statistical properties of a suspect image with the theoretically expected statistical properties of its carrier counterpart such that it is possible to determine the likelihood that a suspect image is a stegogramme.

Statistical attacks are often preferred to visual attacks and structural attacks because they can be automated. This means that there is less pressure on the steganalyst to determine whether an image is a stegogramme or not because the computed result essentially does this on its own. Computed decisions should greatly reduce the total frequency of false-negative results as they are not prone to personal interpretation as visual attacks are. Another benefit of statistical attacks is that they do not require a deep knowledge of what the cover image should look like. Whilst for structural attacks this was a very big part of the success, statistical attacks simply form an analysis based on what is presented from the suspect image alone. However, a deep knowledge of various embedding algorithms is important. If the steganalyst knows a wide selection of embedding tools then they can better design a statistical attack that identifies the artifacts of their embedding process.

3.2 Blind Steganalysis

Blind steganalysis on the other hand is a much harder task, and means that the steganalyst has no reason to believe that covert communications is taking place. In this case, a set of algorithms are typically developed in order to check for signs of tampering. If some signs of tampering are flagged by the algorithms, then it is likely that the suspect file contains steganography.

3.2.1 JPEG Calibration

Perhaps the most important aspect of blind steganalysis is ensuring that we can derive an estimate of the cover image that is as accurate as possible. The attacks that follow this procedure often compare the data in the estimated cover image to that of the suspect image, so it is imperative that the data of the estimate is as sound as possible so as to not obscure the results. One of the most famous approaches for creating an estimate of the cover image is the model proposed by Jessica Fridrich in [5] known as JPEG Calibration. The method takes advantage of the fact that most stego-systems encode the message data in the transform domain during the compression procedure to produce JPEG stegogrammes. Given that the JPEG compression algorithm operates by transforming the image into 8x8 blocks, and it is within these blocks that the encoding of the message operates, we can estimate the cover work by introducing a new block structure and comparing it with that of the suspect image. When there is a large difference, it suggests that the suspect image is a stegogramme, where as little difference typically indicates that the image is innocent The general methodology of the calibration process decompresses the suspect image using its quantization table, removes 4 pixels from each side, and then recompresses the result using the same quantization table. Visually, and technically (by measures such as PSNR6), the calibrated image is still very close to that of the suspect image. However, as a result of cropping the image and recompressing, we effectively break the block structure of the suspect image because the second compression does not consider the first.

When testing the calibration process, it became apparent that the best methodology was to crop the image by 4 pixels in every direction (top, bottom, left, and right). Some literature suggests that 4 pixels should be cropped from the left-hand side of the suspect image, and a further 4 pixels should be cropped from the right-hand side of the image. However, this method does not remove the block structure as well as it should, as this is only equivalent to a half block shift to the side; the block structure from top to bottom remains intact. Cropping from all edges ensures that the entire block structure is removed, and thus a more accurate estimation is derived

3.2.2 Blockiness

Now that we can derive an estimate of the cover image, we need to find some statistical property that differs between the calibrated image and the suspect image such that we can determine the probability that the image is a stegogramme. One of the strongest methods for achieving this is known as Blockiness which takes advantage of the fact that JPEG-driven stego-systems encode the message data in the same 8x8 blocks that are used for compression. The method is defined best by Dongdong Fu in [8] when it is stated that: "[Blockiness] defines the sum of spatial discontinuities along the boundary of all 8x8 blocks of JPEG images".

Essentially, the logic behind Blockiness is that a stegogramme will contain a different set of coefficients across the boundaries of each 8x8 block to that of a clean image. We can therefore total the sums of the boundaries column-wise and row-wise for both a suspect image and a clean image (or our calibrated image) and then calculate the difference between the two. A large difference suggests that the image is a stegogramme, whilst a small difference is probably down to compression, and therefore reflects a clean image. The formula for calculating the Blockiness of an image is shown in equation

$$B = \sum_{i=1}^{\left[\frac{M-1}{8}\right]} \sum_{j=1}^{N} |g_{8i,j} - g_{8i+1,j}| + \sum_{j=1}^{\left[\frac{N-1}{8}\right]} \sum_{i=1}^{M} |g_{i,8j} - g_{i,8j+1}|$$

where gi;j refers to the coordinates of a pixel value in an MxN grayscale image. As we can see from the above equation, the formula operates in a column-wise and row-wise motion rather than calculating the blockiness for each 8x8 block individually. This is achieved by firstly calculating the sum of the values for the 8th row, and then calculating the sum for its neighbouring row (row 9). This process is then repeated for every row-wise multiple of 8, where the each sum is added to the accumulated total until the sums of all the rows have been calculated. The same method is then instantiated for the columns, before finally adding the two totals together. This value is the Blockiness of the image.

4. CONCLUSION

Here in this paper we have discussed various steganalysis schemes for breaking steganogrhy. We have been able to see the strengths and weaknesses of various stego-systems, not only from a steganographic viewpoint, but also in terms of how easy the artifacts of embedding can be spotted via steganalysis. By researching both sides of the field in parallel, it has been interesting to note that a trade-off seems to exist. It seems to be the case that the easiest stego-systems to implement, are also the easiest to attack, whereas the more complicated stego-systems are much harder to attack. This of course makes perfect sense as the more complex systems are likely to be so because they embed the message data in a more intricate fashion than the simpler systems.

5. REFERENCES

[1] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. "Digital Watermarking and Steganography (Second Edition)", Morgan Kaufmann Publishers, ISBN: 978-0-12- 372585-1, 2007.

[2] A. Dennis and B. Wixom. "Systems Analysis & Design (Second Edition)", John Wiley & Sons, Inc., ISBN: 04-7136815-6, 2003.

[3] S. Dumitrescu, X. Wu, and Z. Wang. "Detection of LSB Steganography via Sample Pair Analysis", Lecture Notes in Computer Science, vol. 2578, pp. 355-372, 2003.

[4] H. Farid. "Detecting Hidden Messages Using Higher-Order Statistical Models", Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 2002.

[5] J. Fridrich, M. Goljan, and D. Hogea. "Attacking the OutGuess", Proceedings of the 3rd Information Hiding Workshop on Multimedia and Security 2002, Juan-les-Pins, France, 2002.

[6] J. Fridrich, M. Goljan, and D. Hogea. "Steganalysis of JPEG Images: Breaking the F5 Algorithm", Lecture Notes in Computer Science, vol. 2578, pp. 310-323, 2003.

[7] J. Fridrich. "Feature-Based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes", Lecture Notes in Computer Science, vol. 3200, pp. 67-81, 2004.

[8] D. Fu, Y. Shi, D. Zou, and G. Xuan. "JPEG Steganalysis Using Empirical Transition Matrix in Block DCT Domain", IEEE: 8th Workshop on Multimedia Signal Processing 2006, pp. 310-313, 2006.

[9] R. Gonzales, R. Woods, and S. Eddins. "Digital Image Processing Using MATLAB", Publishing House of Electronics Industry, ISBN: 7-5053-9876-8, 2004.

[10] N. Johnson and S. Jajodia. "Exploring Steganography: Seeing the Unseen", IEEE Computer, vol. 31, no. 2, pp. 26-34, 1998.

[11] N. Johnson and S. Katzenbeisser "A Survey of steganographic techniques", Information Hiding, Artech House, pp. 43-78, 2000.

[12] A. Ker. "Improved Detection of LSB Steganography in Grayscale Images", Lecture Notes in Computer Science, vol. 3200, pp. 97-115, 2005. 117 REFERENCES

[13] K. Lee, A. Westfeld, and S. Lee. "Category Attack for LSB Steganalysis of JPEG Images", Lecture Notes in Computer Science, vol. 4283, pp. 35-48, 2006.

[14] M. Leivaditis. "Statistical Steganalysis", Master's thesis, Department of Computing, University of Surrey, 2007.

[15] N. Memon, I. Avcibas, and B. Sankur. "Steganalysis Based on Image Quality Metrics", emphIEEE: Security and Watermarking of Multimedia Contents, vol. 4314, 2001.

[16] C. Ming, Z. Ru, N. Xinxin, and Y. Yixian. "Analysis of Current Steganography Tools: Classifications & Features", Intelligent Information Hiding and Multimedia Signal Processing 2006, pp. 384-387, 2006.

[17] N. Provos and P. Honeyman. "Detecting Steganographic Content on the Internet", CITI Technical Report, vol. 1, pp. 1-11, 2001.

[18] N. Provos. "Defending Against Statistical Steganalysis", Proceedings of the 10th USENIX Security Symposium, vol. 10, pp. 323-335, 2001.

[19] N. Provos and P. Honeyman. "Hide and Seek: An Introduction to Steganography",IEEE: Security & Privacy, vol. 1, pp. 32-44, 2003.

[20] X. Quan, H. Zhang, and H. Dou. "Steganalysis for JPEG Images Based on StatisticalFeatures of Stego and Cover Images", Lecture Notes in Computer Science, vol. 4681.