On Two Different Methods for Steganography Detection in JPEG Images with Benford's Law

Panagiotis Andriotis, Theo Tryfonas, George Oikonomou, Theodoros Spyridopoulos

p.andriotis@bristol.ac.uk, t.tryfonas@bristol.ac.uk, g.oikonomou@bristol.ac.uk, th.spyridopoulos@bristol.ac.uk

Faculty of Engineering
University of Bristol
Bristol, United Kingdom

Alexandros Zaharis

azaharis@admin.grnet.gr

GRNET S.A.
Athens, Greece

Adamantini Martini

dimart@gmail.com

SIEMENS Enterprise Comms.
Athens, Greece

Ioannis Askoxylakis

asko@ics.forth.gr

Institute of Computer Science
Foundation for Research and Technology-Hellas
Heraclion, Greece

Abstract

The practice of steganography, which in a computer context usually means manipulating multimedia content to embed hidden messages, may be used by criminals worldwide to facilitate their communication instead of, or complementary to, encryption. There is even speculation that global terrorist groups have been using steganography to communicate in covert ways. This paper will introduce steganography and discuss practical aspects of its detection. It will also discuss two recently proposed methods for detecting whether hidden messages exist in JPEG images using Benford's Law. The Law describes the logarithmic distribution of leading digits in sets of naturally set numbers and has been used with success in detecting financial fraud and election rigging in the past. The first approach examines the lead digit distribution of the raw contents of the bytes of a suspect image, whilst the second examines the distribution of lead digits of quantised discrete cosine transform (DCT) coefficients of the JPEG encoding. Both methods produce fast and credible results and are supported by open source toolkits that can be used by law enforcement and investigative authorities worldwide.

Keywords: steganography detection, Benford's Law, computer forensics.

1 Introduction

Data hiding techniques can be performed for various reasons within a system including the facilitation of malware attacks, exchanging secret information via the Internet or hiding data for later use in a compromised environment by an attacker. One of the methods that have been introduced to hide information and covertly spread hidden data without causing suspicion is steganography. A widely used carrier of secret communication is the JPEG image because of the fact that it can be produced by a large range of devices like modern smartphones, tablets, cameras or image processing tools and can be easily exchanged between a wide scope of applications [1]. Steganography aims to transport a message in a covert way by embedding it in a transport medium called a stego-carrier. The carrier with the secret message is known as a stego-cover. To make a steganographic algorithm more powerful, encryption can be used on the secret data, using a stego-key.

Chandramouli et al. [2], define steganalysis as the art of attacking and breaking steganographic methods, aiming either to detect the presence of a secret message in a medium or to extract it. Universal (or blind) steganalytic methods are designed to determine whether an object is carrying a hidden message, without prior knowledge whether the carrier is a stego or the steganographic method that was used to embed a message into the carrier. Steganalysis in a JPEG image is basically categorized on two approaches: visual attacks and statistical attacks. The former usually demand long training steps and a variant amount of resources. The latter attacks are considered to be more efficient and consequently several can be found in the literature [3]. They are based on the fact that after applying steganographic methods on images, the histograms and other high order statistics are usually modified. Postmodern blind steganalytic schemes engage supervised learning to distinguish the plain

media and the stego images and also recognize the data hiding algorithm that was used to embed the secret message [4].

The economic growth and technological bloom led to significant increase of storage ability and therefore there is a growing need for forensic investigators to analyze large volumes of images to detect possible hidden evidence. Benford's empirical law of anomalous numbers [5] that has been successfully used for fraud detection in numerous areas can be also applied to perform various forensic tasks on grey scale [6] and colour JPEG images. Numerous tools have been developed to automate the process of locating suspect carrier files of different file types using visual or statistic analysis attacks. Most of these work against specific steganography algorithms and are usually time consuming.

In this paper we demonstrate our steganalysis methods (already published in literature) using generalized forms of the Benford's Law [7], [8]. The first method focuses on the reconstruction of a prototype JPEG image [9] and the comparison of it against the suspicious potential stego-carrier file. We present the procedures of the creation of the reconstructed image, resembling the data structure of the original image and also the design and usage of a lightweight forensic tool utilizing the aforementioned technique in order to blindly detect image stego-carriers. The second algorithm is an attack that performs steganalysis in a fast and accurate fashion and presents satisfactory detection and false positive rates. We analyze the quantized coefficients of colour images and we predict the behaviour of the distributions of their leading digits. The deviations of these distributions can then be considered as an indication that the examined image is a stego. Furthermore, we developed two automated tools implementing the attacks that can be used to apply blind steganalysis and assist forensic analysts to identify suspicious JPEG images. Both methods and tools perform quite well and are expandable in different file formats. We assess their performance using various images from widely used image databases and our own set created by the use of a smartphone. Our analysis illustrates comparative evaluation with open source steganalysis software like Stegdetect, Camouflage and Invisible Secrets.

The rest of this paper is organized as follows. In Section 2, we refer to prior art in the field and discuss some theoretical background. In Section 3 we demonstrate our detection algorithms. The experimental results are given in Section 4 and in Section 5 we evaluate the tools by comparing them with popular steganalytic programs. The conclusions are drawn in Section 6.

2 Prior Work and Literature Review

2.1 JPEG Image Steganalysis

JPEG compression is a lossy algorithm that aims to discard information imperceptible to human eye while leaving unchanged the general details of the image while reducing its data size. A detailed recitation of the compression procedure of a data stream with the JPEG standard can be found in [10]. It includes the conversion of the pixel colours representation from RGB (Red, Green, Blue) to YC_bC_r, the downsampling of the chrominance values (usually by a factor of two), the transformation of values to frequencies (using 8x8 pixel blocks), the quantization process and zigzag ordering, ending with lossless compression using a variant of Huffman encoding. An image consists of pixels and each pixel usually has three bytes representing its three basic colour components: Red, Green and Blue. Thus, the first step to the JPEG encoding procedure is to convert these pixel values from RGB to YC_bC_r, which is another colour space with three components. Y represents the brightness of an image (called luminance) while C_b and C_r represent colours (called chrominance). The human eye can recognize more easily the difference in the luminance of an image than the chrominance coefficients [11]. The quantization process is done using type II DCT transformation. It is a mathematical function (uses cosine functions) that converts the pixel values of 8x8 blocks to blocks of 64 frequency coefficients.

A JPEG image can be a very convenient cover medium because it usually contains large amounts of space where one can embed data. McBride et al., [1] mention that there are various factors that result in a successful steganographic attempt such as the cover image characteristics and the embedding technique. A general belief is that the image should lack large areas of similarities. Among the popular techniques already used for data hiding in images is the Least Significant Bit (LSB), the DCT encoding and the Fuse algorithm. The embedding techniques that use the quantized DCT coefficients of the image usually hide data by applying LSB encoding in positive integers coefficients. McBride et al. [1] offer a list of tools that use the quantized DCT coefficients to embed information in JPEG images stating that the procedures following the quantization phase are lossless, thus the hidden information can then be obtained. Indicative algorithms from this category are JSteg, Outguess,

JPHide, in contrast to Camouflage or Invisible Secrets that use the Fuse method. The former algorithms introduce irregularities in the statistics of the quantized DCT coefficients of a colour JPEG image and our methods aim to reliably detect such irregularities.

Modern steganalysis deploys machine learning techniques, which are based on image features that get altered during the embedding process, utilising support vector machines or ensemble classifiers [12]. The features constitute a model for pure images that can be used against the suspected stego carriers. Despite their remarkable accuracy these techniques are time consuming, introducing complex concepts and extensive training steps. We, therefore, deploy steganographic models based on Benford's Law providing lightweight statistical attacks that achieve satisfactory precision levels. By using statistical attacks we determine whether the examined data comply with specific rules that normal images would probably follow. For example, the Chi-squared test compares the statistical behaviour of a suspected image with the theoretically expected properties of its carrier [13]. Histogram attacks can also be classified as statistical. They generally depict deviations in the distribution of the frequencies of DCT coefficients of a JPEG image, revealing the presence of a steganographic action.

2.2 Benford's Law and Generalized Benford's Law

A 19th century note by Newcomb (1881) [14] presents the first attempt to explain the behavioural pattern of the significant digits in a set of natural numbers, listing the probabilities of their occurrence. Benford [5] restated the law fifty years later. He observed large groups of natural numbers and concluded that in a set of natural numbers, the probability of the first digit x of a number being 1 is higher than that of the first digit being 9.

$$P(x = 1) > P(x = 2) > \ldots > P(x = 8) > P(x = 9)$$

He also noted that the distributions of the appearance of the leading digits in a set of natural numbers follow a specific logarithmic rule, which was named Benford's Law.

$$p(n) = \log\left(1 + \frac{1}{n}\right), \quad n = 1, 2, \ldots, 9 \quad (1) \qquad \textit{(Benford's Law)}$$

(p(n) is the probability of n being the significant digit of a number in a set of natural numbers.)

Sets should contain as many random numbers as possible. The law is applicable to different groups of natural numbers such as population, addresses and death rates and we can conclude that in a set of natural numbers, it is more possible to find numbers with the significant digit to be 1 than 8 or 9. Schäfer et al., [15] detected fraud and fake survey results at the past using the Benford's Law. The basic principle of this detecting scheme is that natural data generally follow the first digit law in contrast to maliciously altered data.

Image processing and digital forensics related literature also features the logarithmic rule ([6], [16]). Fu et al. [6] studied the behaviour of the JPEG image block coefficients before and after the quantization process. In their experiments they consider only 8 bit grey scale pictures, using as main reference the widely used database of TIFF images called Uncompressed Colour Image Database (UCID) [17]. The experiments on this set of images (after using JPEG compression with a variety of quality factors) show that the distributions of the quantized DCT coefficients follow a logarithmic trend similar to the Benford's Law, thus, the authors propose a new generalized model to describe their behaviour.

$$p(n) = N \cdot \log\left(1 + \frac{1}{s + n^q}\right), \quad n = 1, 2, \ldots, 9 \quad (2) \qquad (\textit{Generalized Benford's Law:" gBL"})$$

(N, s and q are parameters balancing those distributions under different compression quality factors.)

We note that equation (2) [6], is equal to the Benford's Law equation (1) when N = 1, s = 0, q = 1.

3 Algorithms and Structures

In this section we will demonstrate the principles behind our steganographic methods. The first algorithm reconstructs the given image and compares the file structure of both images. The second approach attacks the internal image structure comparing the distributions of the quantized DCT coefficients with the expected deviations of an image with the same characteristics. Both techniques are based on the Benford's Law.

3.1 Image Reconstruction and First - Least Digit Distribution Steganalysis

Before demonstrating this method, we present the four image file types we are using: The *original file***,** which would be a JPEG image created particularly with Microsoft Paint and the *carrier file* which will be the stego after the steganography application on the original file with either JPHSWin, Camouflage or Invisible Secrets. Also, we are using the *reconstructed original file***,** which will be generated by our tool, 'Ben-4D', in order to simulate the original (MS Paint) JPEG file and finally, the *suspect file***,** which can be either an original or a carrier file. The objective of this method is to generate a reconstructed image file as similar as possible to the original. To achieve this task we must identify the encoding algorithm of the suspect JPEG image reading the EXIF metadata. "JPEGSnoop" is an open source program used to analyze the settings that were used when creating the file. Thus, our tool acts partially as an MS Paint Simulator in an attempt to reconstruct an almost identical original file with the given suspect file.

Steganography on a JPEG image can cause alteration in the byte array structure of the file. We estimated that it is critical to detect changes on the first and least digit (f.l.d.) of each byte, as those are mostly affected by popular methods of steganography while differences on the rest of the digits are of minor importance and can be safely discarded. We are using this general form of Benford's Law (G.B.L.) in a method to detect deliberate alterations that may have been caused by the steganography application. Figure 1 describes the procedure.

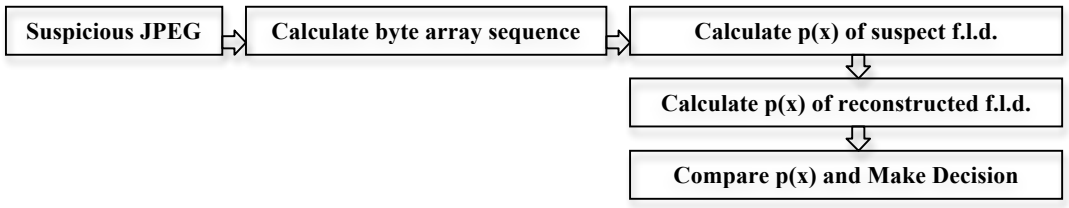

Figure 1: 'Ben-4D' Design Concept.

So, if we have a JPEG image <S> we will first calculate its byte array sequence and then the probability p(x) of the first and the last digit of the byte sequence. Then we will estimate the probability p(x) of the first and the last digit of the byte array sequence for the reconstructed 'original' file and finally we will compare the results and using a predefined threshold we will make a decision whether the suspicious file is a stego-carrier or not. For the reconstruction phase we need to obtain the Quantization Table (Quality Factor) and the Software Signature that created the suspicious picture. This task can be easily accomplished by using JPEGSnoop.

The steganography tools we examined (JPHSWin, Camouflage, Invisible Secrets) do not re-encode the JPEG image. This means that some parts of the image are altered during data hiding but not the Software Signature. At the next step we try to simulate the software JPEG encoding system (in our case MS Paint) that was previously extracted. The missing part to do that is to gain the actual source RGB values that will be used as an input to our JPEG encoding system. The Java Advance Imaging API (JAI) is a library that can successfully achieve this task. Using those data as an input to our JPEG encoding system a reconstructed 'original' JPEG image <P> will be created.

Finally, the detection algorithm consists of the following steps.

1. G.B.L. is applied on image <S> and image size and hash value are calculated.
2. G.B.L. is applied on image <P> and image size and hash value are calculated.
3. Results of Steps 1, 2 are compared updating "The Similarity Factor".

The Similarity Factor is a parameter that indicates how similar two files are. Its values range from 0 to 9, which is the highest similarity degree and is calculated by the different algorithms depending on whether LSB or Fuse steganography was used [7]. We examined a large JPEG image files set and deduced that the Similarity Threshold for MS Paint has approximately a value ST = 5. So, the decision making model is that if ST >= 5 then the examined image does not contain any hidden messages. If ST < 5 steganography might be present. As expected, if the tool attacks against known fingerprints the steganographic algorithms leave when they hide data, it will become more strict and successful. Such fingerprints include the following: JPHSWin uses no standard Huffman tables, Camouflage and Invisible Secrets lead to noticeable differences in size between carrier and reconstructed file, Invisible Secrets does not use standard headers, 'Camouflaged' images are characterized by non proprietary file termination and, generally, altered bits follow specific patterns.

3.2 Quantized DCT coefficients based Steganalysis

Previously, in Section 2.2 we noted that the gBL was proposed by Fu et al. [6] but only the luminance component of each image was taken into consideration during analysis. Thus, we investigated the behaviour of the leading digits of the quantized DCT coefficients of colour JPEG image components, luminance and chrominance. By doing that we extended previous results and created a new reference to describe the expected distributions of the quantized DCT coefficients of a JPEG image. Our steganographic tool 'StegBennie' targets the distributions of the leading digits (fd) that can be extracted from the quantized coefficients of JPEG images. Furthermore, it is important to know the compression quality used to create the image. This quality factor can be easily found by looking at the metadata stored in the header of the image. The tool uses the standard luminance and chrominance quantization tables, provided by the Independent JPEG Group (IJG) and its structure is described in the following Figure 2.

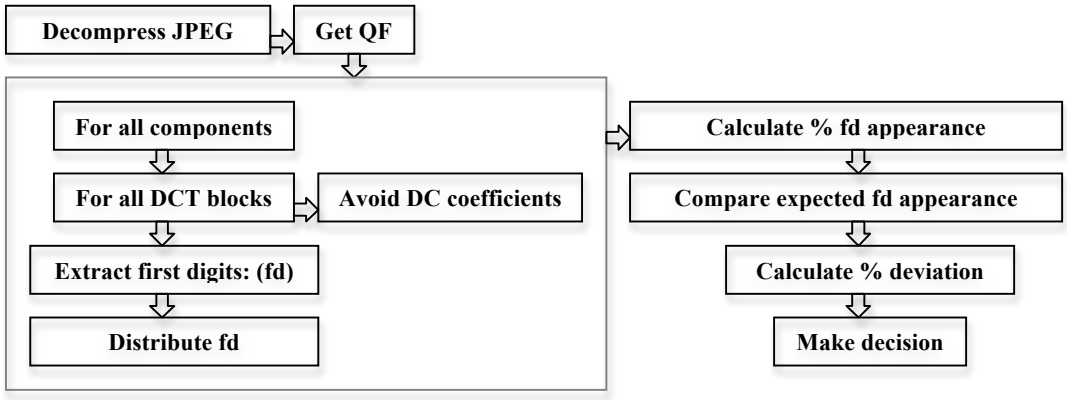

Figure 2: 'StegBennie' Design Concept.

After decompressing the image we read the metadata and find the compression quality factor. We then calculate the appearance of the first digits (fd) of the quantized DCT coefficients of all the components of the colour JPEG image. We are looking at the DCT blocks (8x8) that constitute the image and extract the first digit of each coefficient. For example, if the first row of an 8x8 block of coefficients is [211 22 12 6 1 0 0 0], the first digits are [x 2 1 6 1 x x x] (211 is the DC coefficient and it is excluded and also the zeros are not taken into consideration). This process leads to the calculation of the % percentage of appearance of each leading digit. Then we estimate the first digits expected distribution (given by equation (2)) and finally compare the deviations between the expected and the calculated distributions. The deviation between the expected and the estimated distributions will help to decide if the image is a stego or not.

The method needs a model that will describe the expected distributions of the first digits in case we a-priori know the compression quality factor of the colour image. This model is given by equation 2 and we proved that it is still a reliable metric to describe the first digits distributions of colour JPEGs. To achieve that, we used the second version of the UCID, which contains 1338 uncompressed TIFF images. We made seven folders containing 7x1338 JPEG images that had never been compressed before. Matlab compressed the TIFFs with different quality factors using functions like 'imread' and 'imwrite'. Afterwards, we estimated the distributions of the first digits of the quantized DCT coefficients and the mean distributions for each digit.

We used the Matlab's Curve Fitting Toolbox to calculate the goodness-of-fit of the generalized Benford's Law with new parameters N, s, q and Matlab shows that the gBL (equation 2) describes the mean distributions perfectly (R-Square = 1, Adjusted R-square = 1). Table 1 presents the values of parameters N, q, s for each quality factor and is the model we are using to describe the expected distributions of the leading digits of the suspicious image with the calculated by StegBennie.

In our experiments, we measured the impact of steganography on the first digits distributions. We chose random images from our seven folders and embedded data to them using JPHide, Outguess and Vsl. Then, we calculated the deviations of the distributions of the first digits of the quantized DCT coefficients for each stego-carrier. This process gave us a clear picture of the consequences these algorithms cause to the distributions of the first digits. We examined the deviations of the distributions of the significant digits of the quantized coefficients of pure

images and about 480 stego-carriers compressed with different quality factors. Figure 3 illustrates deviations of first digit distributions for images compressed with quality factor 75. The blue line indicates deviations that emerged from the examination of pure images and the red line indicates the deviations for the same images after applying steganography on them. The horizontal axis of the preceding figures represents the distinct number of images we investigated and the vertical axis states the percentage (%) of the deviations of the distributions of the examined digit. Here we show only data gathered for digits 1, 2, 3 and for compression quality 75.

Quality Factor	***Model Parameters***			***Goodness-of-fit (SSE)***
	N	***q***	***s***	
100	1.608	1.605	0.0702	5.129e-06
90	1.25	1.585	-0.405	7.235e-07
80	1.344	1.685	-0.376	3.007e-06
75	1.396	1.731	-0.3549	3.986e-06
70	1.434	1.766	-0.339	4.455e-06
60	1.514	1.843	-0.3114	5.464e-06
50	1.584	1.909	-0.2875	5.119e-06

Table 1. Behavioural model of gBL for various quality factors.

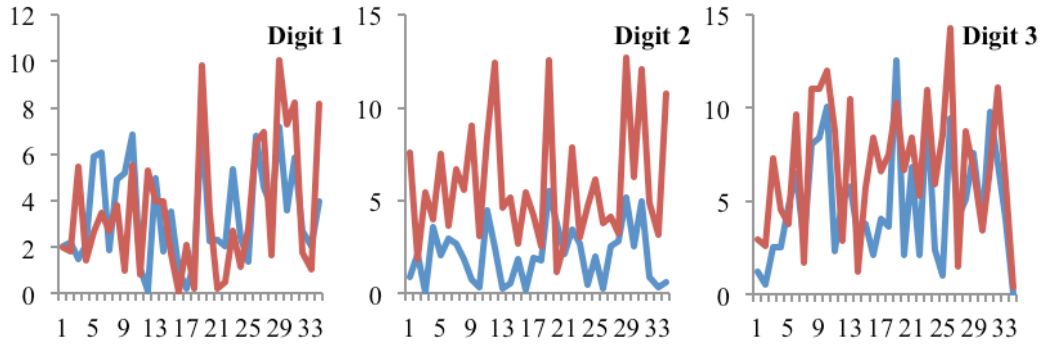

Figure 3: Deviations of first digits distributions for pure and stego images

The overall view we formed from the examination of all digits under numerous quality factors generally follows a specific routine. In more details, the difference in deviations of the distributions of pure images and their respective stego-carries are in most cases larger than 5%. Also, digit 2 appears to have a characteristic that no other digit seems to have. In pure images the deviations of digit 2 are very stable. The range of these deviations is quite convenient and usually varies from 0 to 3 or 4%. Except from that, the deviations of digit 2 after the embedding of a message on the same images behaves similarly, but the deviations exceed a critical threshold (4%). We don't observe the same attitude from digit 1 for example (Figure 3). Here, the deviations are within a small range but we can see that the two lines do not have the same behaviour compared to the two lines of digit 2 where the blue line is almost always below the red line. Thus, in most of the cases, we expect that an image containing hidden data will present deviations higher than a certain threshold T. In the specific example, a suitable threshold should be T = 3. The findings of our experiments demonstrated that if this deviation of digit 2 exceeds a specific threshold, which depends on the quality factor of the compression of the examined image, we can conclude that the image is suspicious. After repeating the same tests to JPEG images using Outguess and Vsl as the embedding algorithms and analyzing data using the same methodology, we discovered that the impact of steganography on the distributions of the first digits was significant. We also confirmed that the deviations of digit 2 were smooth and the aforementioned threshold was sufficient and capable to detect a suspicious JPEG image.

Hiding data with Outguess and Vsl leaves detectable fingerprints. Outguess always uses the quality factor of 75 and Vsl always quantizes with QF = 100 while embedding messages. Consequently, the expected distributions of the examined images are significantly different than the observed. We use these fingerprints at the decision making module of 'StegBennie' and conclude that when we investigate an image which has a quality factor of 75 or 100 and the deviation of digit 2 is really large, we deduce that Outguess or Vsl was used, respectively.

4 Experimental Results

In this section we concatenate in Tables the findings of our experiments when we tested the tools against stego

and non stego-carriers. Both tools and methods function successfully and the detection rates are satisfactory against the algorithms we already mentioned in former sections. Before testing 'Ben-4D', stego-carrier files were divided into three different groups of five hundred original files each. For each group JPHSWin, Camouflage and Invisible Secrets were used to embed the smallest in size, file possible which was an ASCII txt file (1Kb). Table 2 shows that the false positive rate was about 1.5% but this percentage will decrease when the hidden file is larger (5Kb).

JPEG Image Dimensions	320x240				600x320				800x600			
File Type	*Original*	*JPHSWin*	*Camouflage*	*Invisible Secrets*	*Original*	*JPHSWin*	*Camouflage*	*Invisible Secrets*	*Original*	*JPHSWin*	*Camouflage*	*Invisible Secrets*
Number of Files	500	500	500	500	500	500	500	500	500	500	500	500
Average Size (Kb)	30	34	31	31	100	104	101	101	200	205	201	201
G.B.L. Hit Rate (%)	89	89.1	99.6	99.7	88.5	86.7	99.6	99.7	88	82.2	99.6	99.7
False Positive Stego Detection	10%				15%				20%			
Scan Time (sec) per Item	~1	~2	~1	~1	~1	~2	~1	~1	~1	~2	~1	~1
Total Time (min)	8.3	16.6	8.3	8.3	8.3	16.6	8.3	8.3	8.3	16.6	8.3	8.3
G.B.L. + Signature Hit Rate (%)	99.9	99.8	100	100	99.9	98.1	100	100	99.9	97.4	100	100
False Positive Stego Detection	0.1%				0.1%				0.1%			
Scan Time (sec) per Item	~2	~3	~2	~2	~2	~3	~2	~2	~3	~4	~3	~3
Total Time (min)	16.6	25	16.6	16.6	16.6	25	16.6	16.6	25	33.3	25	25

Table 2: Ben-4D's hit rates and false positives (hidden data: 1Kb)

Table 2 presents the hit rates achieved by the G.B.L. detection algorithm in comparison to the combination of G.B.L. and utilization of detection fingerprints. It also demonstrates that the larger in size the image is, the smaller the hit rate becomes. This finding can be explained because the overall byte array structure alteration is statistically smaller in large areas, thus more difficult to detect.

We tested the accuracy of the second tool 'StegBennie' in three stages. First, we calculated the algorithm's false positive rate (FPR) (9,370 images) and then we tested the validity of the method on the training set (about 200 images) and finally on a set of images taken by a smartphone (approximately 150 images). Table 3 demonstrates the FPR and the hit rates for the sample of the training set and in a similar fashion, Table 4 states the FPR and the hit rates the tool achieved for the set of the smartphone images.

QF	*F.P.R. (%)*	*JPHSWIN (%)*	*Outguess (%)*	*Vsl (%)*
50	24.47	76.47	100	100
60	24.47	73.53	100	100
70	2.94	82.35	80	100
75	20.59	85.29	20	100
80	29.41	73.53	100	80
90	11.76	67.65	100	100

Table 3: Hit rates and false positives of 'StegBennie' on the training data.

QF	*Resolution*	*F.P.R. (%)*	*JPHSWin (%)*	*Outguess (%)*	*Vsl (%)*
Normal QF = 70	small	11.11	88.89	77.78	100.0
	1Mp	0	90.0	75.0	100.0
	3Mp	11.11	55.55	75.0	100.0
	5Mp	12.5	33.33	87.5	100.0
	wide1Mp	0	66.67	50.0	100.0
High QF = 80	small	10.0	100.0	100.0	100.0
	1Mp	11.11	66.67	100.0	100.0
	3Mp	20.0	50.0	100.0	100.0
	5Mp	37.5	50.0	71.43	100.0
	wide1Mp	0	60.0	100.0	100.0
Fine QF = 90	small	30	100.0	100.0	100.0
	1Mp	15.79	66.67	90.0	100.0
	3Mp	44.44	55.55	87.5	100.0
	5Mp	30.0	40.0	100.0	100.0
	wide1Mp	27.27	72.73	90.0	100.0

Table 4: Hit rates and false positives of 'StegBennie' on the smartphone dataset.

5 Evaluation and Comparison with Similar Tools

The demonstrated tools can be boldly compared with other similar programs that utilize different steganalytic algorithms. The achieved detection rates are fair and in some cases better that the respective rates of StegDetect or StegSpy. In Table 5 we can see that 'Ben-4D' has similar success rates with the other tested steganalytic algorithms. Figure 4 shows that 'StegBennie' achieves better results when tested against 'StegDetect'. Furthermore, 'StegBennie' achieved these detection rates faster that the other tool. To be more precise our tool needs half the time that 'StegDetect' spends to perform steganalysis.

File Type	*Original*	*JPHSWin*	*Camouflage*	*Invisible Secrets*
Number of files	1500	1500	1500	1500
Detection Rates (%)				
Ben-4D (full)	99	98	100	100
StegDetect	99.5	99.2	100	100
StegSpy	98	99	100	100

Table 5: Hit Rates Comparison among 'Ben-4D' and other tools.

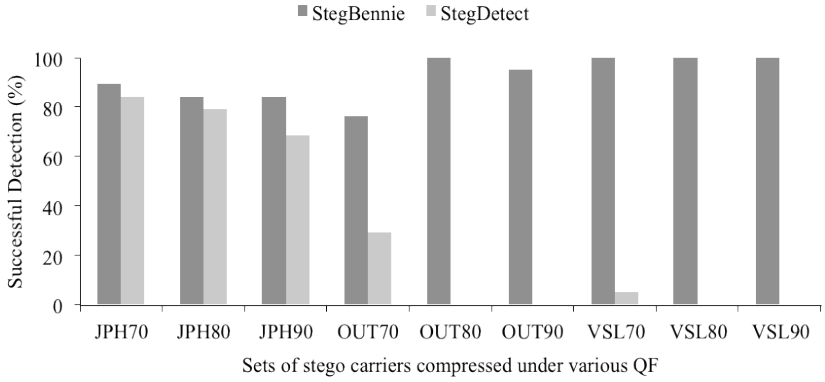

Figure 4: Hit rates Comparison between 'StegBennie' and StegDetect.

Taking all these findings into consideration we can say that the two steganalytic tools we bring to the Forensic community are reliable, fast, easy to use and develop further and will give the opportunity to analysts to perform steganalysis to large volumes of images following an accurate manner and achieving well perceived results. Our proposed use of these methods is as follows: Ben-4D is a tool to be used when images are retrieved through carving and where the metadata of the pictures in question may be missing or unreliable for whatever reason, e.g. examining partially recovered images for steganography. On the other, hand trawling through large sets of images is more suited to the image-processing based approach of StegBennie.

6 Conclusions and Future Work

Benford's Law is a powerful empirical rule giving the chance to analysts to screen suspicious JPEG images and decide if steganography was applied to them. We contribute to the area of steganography detection using two variations of the Law and developing tools that utilize novel algorithms acting fast and accurately. Our goal is to bypass complex learning steps and techniques and use a law abiding by numerous natural sets of numbers. Our steganalytic methods present fair results compared to prior art and are significantly faster. The first algorithm studies the byte array sequence of the image and the second is focused on the internal structure of it. We therefore provide the opportunity to a forensic analyst to decide which of these tools would be more accurate for a given problem or use them together.

However, improvements can be done to both tools. It is essential to develop software that not only will effectively screen JPEG images but it will also be able to detect steganography on any possible image carrier file, such as png, bmp. For this reason we must identify the potentiality of steganalysis with the Benford's Law for popular image types that can be found basically in the Internet. Considering 'Ben-4D', data loss during the reconstruction process can be diminished using enhanced algorithms or lossless transcoding. For 'StegBennie', the importance of payload of the embedded data should be examined to form a stable view about the efficiency of the tool. Also, the decision making model can be enhanced by other factors like the behaviour of the first digits of the DCT blocks before the quantization phase. Finally, image steganography can be used in malware and especially in malicious applications for Android smartphones through third party markets. Image

steganalysis with our tools can control malware spreading because they are lightweight and their code can be migrated in order to create simple steganalytic smartphone applications.

Acknowledgements

This work has been supported in part by the European Union's Prevention of and Fight against Crime Programme "Illegal Use of Internet" – ISEC 2010 Action Grants, grant ref. HOME/2010/ISEC/AG/INT-002. Full details about the two approaches presented here can be found in the authors' previous works in [7,8].

References

[1] McBride B, Peterson G, Gustafson S. A new blind method for detecting novel steganography. In: Digital Investigation 2005; 2(1): 50–70.

[2] Chandramouli R, Kharrazi M, Memon N. Image steganography and steganalysis concepts and practice. 2nd International Workshop on Digital Watermarking (IWDW 2003). Lecture Notes in Computer Science, Vol. 2939; 2004. p. 35–49.

[3] Chandramouli R, Subbalakshmi K. Current trends in steganalysis: a critical survey. In: Proc. 8th International Conference on Control, Automation, Robotics and Vision (ICARCV 2004), Vols. 1–3; 2004. p. 964–7.

[4] Solanki K, Sarkar A, Manjunath BS. YASS: yet another steganographic scheme that resists blind steganalysis. In: Furon T, editor. Information hiding. LNCS, Vol. 4567; 2007. p. 16–31.

[5] Benford F. The law of anomalous numbers. In: Proc. of American philosophical society; 1938. p. 551–72.

[6] Fu D, Shi YQ, Wei S. A generalized Benford's law for JPEG coefficients and its applications in image forensics. Proc. 9th conference on security, steganography, and watermarking of multimedia contents. In: Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), Vol. 6505; 2007: 65051L–65051L–11.

[7] Zaharis A, Martini A, Tryfonas T, Ilioudis C, Pangalos G. Lightweight Steganalysis Based on Image Reconstruction and Lead Digit Distribution Analysis. In: International Journal of Digital Crime and Forensics 2011, 3(4): 29-41.

[8] Andriotis P, Oikonomou G, Tryfonas T. JPEG steganography detection with Benford's Law. In: Digital Investigation 2013; 9(3-4): 246–257.

[9] Nosratinia, A. Enhancement of JPEG-Compressed Images by Re-application of JPEG. In: Journal of VLSI Signal Processing Systems For Signal Image and Video Technology (2001); 27(1-2); 69–80

[10] Wallace G. The JPEG still picture compression standard. In: IEEE Transactions on Consumer Electronics 1992; 38(1): R18–34.

[11] Lee K, Westfeld A, Lee. Category attack for LSB steganalysis of JPEG images. In: Digital watermarking (5th international workshop) IWDW 2006, Korea, LNCS, Vol. 4283. Springer-Verlag; p. 35–48.

[12] Zong H, Liu FL, Luo XY. Blind image steganalysis based on wavelet coefficient correlation. In: Digital Investigation 2012; 9(1): 58–68.

[13] Westfeld A, Pfitzmann A. Attacks on steganographic systems–breaking the steganographic utilities EzStego, Jsteg, Steganos, and S-Tools and some lessons learned. In: Proc. 3rd international workshop on Information Hiding (IH 99) Vol. 1768. Springer Berlin/Heidelberg; 2000. p. 61–76.

[14] Newcomb S. Note on the frequency of use of the different digits in natural numbers. American Journal of Mathematics 1881; 4(1): 39–40.

[15] Schäfer C, Schräpler JP, Müller KR, Wagner GG. Automatic identification of faked and fraudulent interviews in surveys by two different methods. Deutsches Institut für Wirtschaftsfor-schung; 2004.

[16] Pérez-González F, Heileman GL, Abdallah CT. Benford's law in image processing. In: Proc. IEEE International Conference on Image Processing, (ICIP 2007); 2007. p. 405–8.

[17] Schaefer G, Stich M. UCID–an uncompressed colour image database. In: Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), Vol. 5307; 2004. p. 472–80.