Digital Investigation 9 (2013) 246–257

Contents lists available at SciVerse ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

JPEG steganography detection with Benford's Law

Panagiotis Andriotis*, George Oikonomou, Theo Tryfonas

Crypto Group, University of Bristol, Faculty of Engineering, Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK

ARTICLE INFO

Article history:
Received 29 October 2012
Received in revised form 23 January 2013
Accepted 25 January 2013

Keywords:
Steganalysis
Generalized Benford's Law
Steganography detection
Data hiding
Quantized DCT coefficients

ABSTRACT

In this paper we present a novel approach to the problem of steganography detection in JPEG images by applying a statistical attack. The method is based on the empirical Benford's Law and, more specifically, on its generalized form. We prove and extend the validity of the logarithmic rule in colour images and introduce a blind steganographic method which can flag a file as a suspicious stego-carrier. The proposed method achieves very high accuracy and speed and is based on the distributions of the first digits of the quantized Discrete Cosine Transform coefficients present in JPEGs. In order to validate and evaluate our algorithm, we developed steganographic tools which are able to analyse image files and we subsequently applied them on the popular Uncompressed Colour Image Database. Furthermore, we demonstrate that not only can our method detect steganography but, if certain criteria are met, it can also reveal which steganographic algorithm was used to embed data in a JPEG file.

1. Introduction

The use of several means of covert communication is appealing among individuals or groups that are interested in securing the content of an exchange concealing the act of their interactions. Steganography is one of the methods which have been introduced in order to hide information and covertly spread hidden data through public channels without causing suspicion. JPEG images constitute a widely used medium of secret communication, partially thanks to the fact that they can be produced by any camera, smartphone or image processing tool and can be easily exchanged between a variety of applications (McBride et al., 2005).

Steganography aims to transport a message in a hidden fashion by embedding it in a transport medium called a carrier (Fridrich et al., 2001). The grouping of the carrier with the secret message is known as a stego medium or stego cover. The detection of steganographic algorithms and techniques can be a hard task, even more so if the secret data are encrypted with a stego key. Steganalysis is the process of attacking and breaking steganographic methods, either by simply detecting the presence of a secret message or by extracting and potentially destroying it (Chandramouli et al., 2004). The success of a steganalytic method can be quantified either by the accuracy of the prediction of a secret message's presence in a stego object or by the extraction of the hidden information. Steganalysis methods can be further classified into two broad categories: *targeted* and *blind* (or *universal*). In *targeted* steganalysis the attack is mounted against an already known embedding technique. *Blind* steganalysis methods aim to determine whether an object is carrying a hidden message, without any a-priori knowledge.

When the stego carrier is a JPEG image steganalysis is prominently based on two approaches: *visual* and *statistical* attacks (Westfeld and Pfitzmann, 2000; Jolion, 2001). Visual attacks demand long training steps and a significant amount of resources. Statistical attacks are more resource-efficient and as a result, several can be found in the literature (Chandramouli and Subbalakshmi, 2004). These are

* Corresponding author. Tel.: +44 117 33 15740; fax: +44 117 33 15719.
E-mail addresses: p.andriotis@bristol.ac.uk (P. Andriotis), g.oikonomou@bristol.ac.uk (G. Oikonomou), theo.tryfonas@bristol.ac.uk (T. Tryfonas).

1742-2876/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.diin.2013.01.005

based on the fact that the images' histograms or high order statistics get modified after the steganographic techniques take place. Modern blind steganalytic schemes engage supervised learning to differentiate between the plain media and stego images and also distinguish the data hiding algorithm used for steganography (Solanki et al., 2007).

Benford's empirical law of anomalous numbers (Benford, 1938) has been successfully used in the past for fraud detection in the accountancy sector. It has also been demonstrated that the law in a generalized form can be employed to perform a series of forensic tasks on JPEG images, such as the detection of double compression (Fu et al., 2007). This work was limited to grey scale images however. The generalized Benford's Law has been employed for steganalysis elsewhere (Zaharis et al., 2011), but there it was applied on raw byte values and not from an image analysis perspective.

In this context, this paper's contribution is two-fold:

- We adopt the generalized Benford's Law as the basis of a novel statistical attack for blind steganalysis and we provide evidence of its applicability on colour JPEG images.
- We demonstrate that the attack can perform steganalysis very quickly and achieves a satisfactory detection rate.

The proposed attack is based on an analysis of the quantized coefficients of a large amount of colour images. Our method indicates that it is possible to predict the behaviour of the distributions of their significant digits and any disturbances of these distributions can then be considered an indication of the presence of steganography. By studying the deviations of their distributions, we propose a decision making model based on our findings related to the behaviour of digit 2. Moreover, we developed a set of automated tools which implement the attack and can be used to conduct blind steganalysis and thus help forensic analysts to identify suspicious colour JPEG images. In order to validate the method and assess its performance, we used it to analyse files taken from a widely-used database of approximately 1340 images, enriched by our own set created by the use of a smartphone. Our analysis includes comparative evaluation with the open source steganalysis software Stegdetect.

The rest of this paper is organized as follows. In Section 2, we highlight our main motivations and discuss some theoretical background. In Section 3 we present our new detection algorithm. The experimental results are provided in Section 4 and the discussion on the results can be found in Section 5. Finally, in Section 6 we present results from testing our method in various steganalytic tasks. The conclusion is drawn in Section 7.

2. Theoretical background and motivation

The term JPEG comes from the consortium that created the standard (Joint Photographic Experts Group). It is one of the most common formats and it is widely used by all the manufacturers of digital consuming products such as digital cameras. It comes from the need to exchange images through different platforms and applications. The main goal of the JPEG compression is to discard information which is imperceptible to the human eye while leaving unchanged the aesthetic details of the image. Simultaneously, the JPEG compression reduces image data size. A detailed presentation of the procedure followed in order to compress a data stream with the JPEG standard can be found in Wallace (1992). Usually the Discrete Cosine Transform (DCT) encoding procedure consists of six basic steps: Conversion of the representation of colours from RGB (Red, Green, Blue) to YC_bC_r, downsampling of the chrominance values (usually by a factor of two), transformation of values to frequencies (using 8×8 pixel blocks), quantization process, zigzag ordering, lossless compression using a variant of Huffman encoding.

In more detail, an image consists of pixels and each pixel usually has three bytes that represent its three basic colour components: Red, Green and Blue. 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 that has three components. Y represents the brightness of an image and is called luminance while C_b and C_r represent colours and they are called chrominance. It is known that the human eye can recognize the difference in the luminance of an image more easily than the chrominance coefficients (Lee et al., 2006). The type II DCT is responsible for the quantization process. DCT is a mathematical transformation (uses cosine functions) that converts the pixel values of 8×8 blocks to blocks of 64 frequency coefficients. These numbers are critical for our method.

A digital image and especially a JPEG image can be a perfect cover medium because it usually has large amounts of space where one can embed information. There are numerous factors that result in a successful embedding procedure such as the embedding technique and the cover image characteristics (McBride et al., 2005). A general assumption is that the image should be busy, meaning that it should lack large areas of similarities. Popular techniques used to hide information in images are the Least Significant Bit (LSB) and the DCT encoding. Embedding techniques focus on the quantized DCT coefficients and they usually embed data by applying LSB encoding in those coefficients that are not equal to zero. In McBride et al. (2005) we can find a list of tools that use the quantized DCT coefficients to embed data in JPEG images. They rely on the fact that the procedures which follow the quantization phase are lossless and the hidden information can then be obtained. Indicative algorithms from this category are Jsteg, Outguess, JPHide and F5. Those techniques introduce irregularities in the statistics of the quantized DCT coefficients of a colour JPEG image. Our goal is to reliably detect such irregularities.

Statistical attacks aim to determine whether the examined data comply with specific statistical rules that normal image files would follow. A very popular attack is the Chi-squared test which compares the statistical behaviour of a suspected image with the theoretically expected properties of its carrier (Westfeld and Pfitzmann, 2000). Histogram attacks, which can also be classified as statistical, depict disturbances in the distribution of the frequencies of DCT coefficients of a JPEG image. These figures can reveal the

existence of a steganographic attempt. A comprehensive and well informed work on steganalysis trends was published by Chandramouli and Subbalakshmi (2004).

Nowadays, machine learning techniques are common in the field of steganalysis. These techniques are based on image *features* which get altered during the embedding process and machine learning is the de facto standard procedure that deals with them utilising support vector machines (SVM) and lately, ensemble classifiers (Zong et al., 2012; Kodovsky et al., 2012). The features constitute a model for natural, pure images which can be used against the suspected stego carriers. However, despite their accuracy, these techniques are time consuming, they introduce extensive training steps and their complexity is high. For this reason we are implementing a new model based on Benford's Law and we introduce a method in order to identify stego carriers in a fast, simple and efficient manner.

2.1. Benford's empirical law of anomalous numbers

The first attempt to decode the behaviour of the first digits in a set of natural numbers was conducted towards the end of the 19th century by Newcomb (1881). This note presents a table which lists the probabilities of occurrence of the first digits of a set of natural numbers. Numbers cannot be zero and they have more than one digit. Fifty years later Benford (1938) rediscovered and restated the law. He investigated large groups of natural numbers and observed that, in all selected groups, the probability of the first digit x of a number being 1 is higher than that of the first digit being 9. Furthermore, the distribution of the appearance of the first (or significant) digits in a set of natural numbers follows a logarithmic rule. Therefore:

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

The mathematical equation which describes the first digits law is presented in Equation (1):

$$p(n) = \log_{10}\left(1 + \frac{1}{n}\right),\ n = 1, \ldots, 9. \tag{1}$$

$p(n)$ represents the probability of n being the first digit of a number in a set of natural numbers. Sets should contain as many numbers as possible in a random fashion. This empirical law is applicable to different groups of natural numbers such as population, addresses, drainage and death rates.

According to this empirical view we are able to predict that, in a set of natural numbers, it is more probable to find numbers with the significant digit to be 1 than 8 or 9. This law looks like it fights against common sense but it is now widely used in the area of expenses and accounting fraud detection and was also introduced in various social occasions. For instance, Schäfer et al. detected fraud and fake survey results using the Benford's Law (Schäfer et al., 2004). The basic principle behind all examples is that natural data generally follow the first digit law quite well in contrast to maliciously changed or randomly guessed data. Some attempts to utilize the results of the findings of the logarithmic law can also be encountered in literature related to image processing and digital forensics (Jolion, 2001; Fu et al., 2007; Pérez-González et al., 2007).

2.2. Generalized Benford's Law

In 2007, Fu et al. presented a new approach to image forensic analysis using the law of anomalous numbers and studied in depth the behaviour of the JPEG image block coefficients (Fu et al., 2007). In this work there are some conclusions about the validity of Benford's Law in the most significant digits of DCT coefficients (before quantization) of the 8×8 pixel blocks of any grey scale JPEG image; the DC coefficients[1] are excluded from the research. Experiments were conducted considering only 8 bit grey scale pictures, using as main reference a widely used dataset of TIFF images called the *Uncompressed Colour Image Database* (UCID) (Schaefer and Stich, 2004). The use of such a database guaranteed that those images have never before been JPEG compressed. They also examine the distribution of the first digits of the *quantized DCT coefficients* that emerge after the quantization process. After completing the calculation of the appearance of significant digits of the DCT coefficients in this set of images, their mean distribution was obtained. The significant digits of DCT coefficients conform quite well to the Benford's Law, with goodness of fit results confirmed by using x^2 divergence.

By conducting thorough experiments on the same set of images, the authors also calculated the mean distribution of the first digits of the quantized DCT coefficients under different quality compression factors (QF = 100, 90, 80, 70, 60, 50). The results show that the distributions of those coefficients also follow a logarithmic trend. A comparison between the mean distributions that they obtained for each compression quality and the expected Benford's Law distributions revealed that the quantized coefficients do not follow the rule Equation (1) in a very strict way as the DCT coefficients do. However, there is also a logarithmic law behind the distribution of the first digits of the quantized DCT coefficients. The model they proposed is described by the following Equation (2):

$$p(n) = N \cdot \log_{10}\left(1 + \frac{1}{s + n^q}\right),\ x = 1, 2, \ldots, 9 \tag{2}$$

N, s and q are parameters which describe precisely those distributions under different compression quality factors. In the special occasion where $N = 1$, $s = 0$ and $q = 1$ we can conclude that Equation (2) which is called the generalized Benford's Law (gBL) (Fu et al., 2007), is equal to the Benford's Law Equation (1).

3. Method and algorithm

Our method focuses on the distributions of the significant digits which can be extracted from the quantized coefficients of colour JPEG images. The decompression of a JPEG image is exactly the inverted process of what we presented in Section 2. In Section 2.2 we underlined that

[1] The upper left coefficient of each block.

the gBL was proposed by investigating 8 bit grey scale images only by Fu et al. (2007). Thus, only the luminance of each image was taken into consideration. For this reason we decided to investigate the behaviour of the quantized DCT coefficients of all the components of a JPEG image; both luminance and chrominance. The investigation contributes to previous work by extending the results and by creating a new reference as a basis to describe the expected distributions of the quantized DCT coefficients of a colour JPEG image. The knowledge of the compression quality is critical at this phase. The compression quality factor can be revealed by looking at the image's metadata. In our experiments we used the standard luminance and chrominance quantization tables, provided by the Independent JPEG Group (IJG).

The basic steps of our method include the calculation of the appearance of the significant digits of the quantized DCT coefficients of all the components of a colour JPEG image. For example, if the first row of an 8×8 block of coefficients is [154 32 1 19 2 0 0 0], the first digits are [x 3 1 1 2 x x x] (154 is the DC coefficient and it is excluded and also the zeros are not taken into consideration). Then we estimate their expected distribution (given by Equation (2)) and finally compare the deviations between the expected and the calculated distributions. We use this information to decide if the image is suspicious or not. In some cases, the same data can be used to determine exactly which steganography algorithm was used to embed the hidden object. We analysed the behaviour of the digits using specific quality factor compressions: QF = 100, 90, 80, 75, 70, 60, 50.

In order to achieve this, we need a model to represent the distributions of the quantized DCT coefficients of any colour image. This can be feasible if we prove that Equation (2) is still a reliable model that describes the probability of appearance of the first digits of the quantized DCT coefficients of a JPEG image, even if these were collected from all the components of the image; luminance as well as chrominance. We used the second version of the UCID for this experiment which contains 1338 uncompressed TIFF images. A Matlab script was written to compress them with different quality factors. The script used Matlab's functions *imread* and *imwrite* and compressed the images within seconds. As a result, we accumulated seven groups of 1338 JPEG images that had never been compressed before. This step was vital for the accuracy of our work because we were able to know the compression history of each image. Secondly, we calculated the distributions of the first digits of the quantized DCT coefficients. After this step the mean distributions for each digit were calculated by Matlab. The algorithm that was used can be described by the following pseudo code.

```
decompress_image();
for all components {
  for each DCT block {
    consider only AC coefficients;
    extract_first_digits();
    distribute_first_digits();
  }
}
calculate_percentage_of_appearance();
```

We estimated the goodness-of-fit of the generalized Benford's Law using the Matlab's Curve Fitting Toolbox. To avoid the calculation of any complex values from Matlab we had to define the boundaries of parameters N, s, q. The use of the curve fitting toolbox for all quality factors resulted in the conclusion that gBL can describe the distributions of the appearance of quantized DCT coefficient first digits of a colour JPEG image in a very satisfactory manner. As a matter of fact, the statistics that Matlab provides to estimate the fitting results, show that the gBL 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. There is also a column which represents the Sum of Squares Due to Error (SSE). SSE is another fitting statistic that Matlab provides and Table 1 shows that in our case this error is infinitely minor.

We are now able to calculate the expected distributions of the appearance of the quantized DCT coefficients. The idea behind this concept is that given the quantization table of the luminance of a JPEG image, we can obtain the compression quality that was used during encoding. Afterwards, we can calculate the distributions of appearance of the coefficients and compare them with the expected distributions. We will be able to estimate the deviations between the distributions (current and expected) and decide if the JPEG image is suspicious or not. In our research we used the percentage of the deviations because it makes the comparison between first digit distributions more reasonable. For example, digit 9's distribution is always between 1 and 2% and digit 1's distribution can vary from 55 to 60%. Their deviations should be measurable and comparable and this is why we should use the % of deviations as a common measurement system.

Subsequently, we measured the impact of steganography on these distributions. We chose random images from our seven sets (each set contained 1338 JPEG files) and we embedded data with JPHide, Outguess and Vsl. JPHide and Outguess hide data in the quantized DCT coefficients. We then calculated the deviations of the distributions of the first digits of the quantized DCT coefficients for each potential stego carrier. By doing this, we gained a clear picture of the consequences that these algorithms cause to the distributions of the first digits. Table 2 shows the % deviations caused to an image when we embedded a text file with JPHSWIN and their (absolute) difference.

At this stage we tried to verge on the issue of finding a reliable indicator that could safely reveal the suspicious image. We focused our interest on the deviations of the distributions of the first digits of the quantized coefficients

Table 1
Goodness of fit for the gBL model for luminance and chrominance.

Quality factor	N	q	s	Goodness-of-fit (SSE)
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

of pure images and stego carriers. The stego carriers were created by the same pure images but they also contained messages (in .txt format) which were embedded by JPHSWIN. We carefully examined about 480 images compressed with different quality factors. Fig. 1 illustrates deviations of first digit distributions for images compressed with quality factor 75. The solid line indicates deviations that emerged from the inspection of pure images and the dashed line indicates the deviations for the same images after applying steganography on them. The horizontal axis of the preceding figures represents the images we examined and the vertical axis states the percentage (%) of the deviations of the distributions of the examined digit. The overview we got by examining the figures we formed from images that were compressed by various quality factors was similar to what we can see on Fig. 1.

The analysis of the difference in deviations of the distributions of pure images and their respective stego carries reveals that the differences are in most cases larger than 5%. Furthermore, we observe that differences in deviations are more extreme for digits 2, 4, 6 and 8. Fig. 1b further reveals a characteristic of digit 2 that no other digit seems to have. When we examined pure images the deviations of digit 2 were very stable. The range of these deviations was quite convenient and usually varied from 0 to 3 or 4%. Except from that, the deviations of digit 2 after the embedding of a message on the same images behaved in a similar fashion, but this time the deviations exceed the 4% threshold. We cannot see the same attitude from the digit 1 for example (Fig. 1a). 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 Fig. 1b. In Fig. 1b we can see that the solid line is almost always below the dashed line. Thus, in most of the cases, we expect that an image which contains a hidden message will present deviations which are higher than a certain threshold T. In the specific example, a suitable threshold would be $T = 3$. It becomes more interesting if we underline that the thresholds for all examined quality factors vary between 3 and 4. Taking these findings into consideration we concluded that the most stable and reliable indicator for a suspicious image to be revealed is the deviation of digit 2. If this deviation 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. We approximated these values statistically for each compression quality and present them in Table 3.

Table 2
Difference between deviations of distributions in a pure image and a stego carrier.

First digits	Deviations (pure)	Deviations (stego)	|Difference|
1	5.117569	0.947102	4.170467
2	0.678150	9.001642	8.323492
3	8.373005	10.988395	2.61539
4	9.832138	1.039585	8.792553
5	3.874760	4.447051	0.572291
6	9.937180	1.376626	8.560554
7	14.700152	17.820417	3.120265
8	8.818516	1.664183	7.154333
9	14.713667	13.687573	1.026094

We should underline at this point that we did not manage to verify the accordance of the previous results with images that were compressed with QF = 100. As a matter of fact, both deviations and differences between the pure images and stego carriers seem like they do not follow any rule that complies with our findings for the other quality factors. This phenomenon occurs because when compressing with quality factor 100, the quantization tables have a very weak effect on the first digits.

We repeated the same tests to JPEG images using Outguess and Vsl as the embedding algorithms. We analysed the data using the same methodology and discovered that the impact of steganography on the distributions of the first digits was significant. Often the difference between the expected and the given deviations was more than 70%. We also confirmed that the deviations of digit 2 were smooth and the thresholds of Table 3 were sufficient and capable to detect a suspicious JPEG image. A closer look at the effects of the application of steganography with Outguess and Vsl revealed that both algorithms change the image quantization tables when they embed a message into their internal structure. Outguess always uses the quality factor of 75 and Vsl always quantizes with QF = 100. Consequently, the expected distributions of the inspected images are significantly different than the observed ones. Our research revealed that Outguess leaves the quantization table of quality 75 as a fingerprint or signature. The same goes for Vsl which turns the quality factor of the stego carriers to 100. In other words, the metadata of a stego carrier created by Outguess or Vsl will always indicate that the quality factor used to quantize the block coefficients is QF = 75 or QF = 100, respectively. We used these fingerprints when we built the decision making module of our programs. If we try to investigate an image which has a quality factor of 75 or 100 and the deviation of digit 2 is really large, we can deduce that Outguess or Vsl was used, respectively.

The research on the behaviour of the first digits of the quantized DCT coefficients of colour JPEG images and the analysis of the data we gathered from their distributions and deviations resulted in the development of a new universal steganalytic tool which we called *StegBennie*. This tool uses the characteristics of the distributions of digit 2 and it is a new approach to the problem of steganalysis of colour JPEG images. StegBennie applies a statistical attack on a JPEG image using the generalized Benford's Law and estimates whether it is a suspicious image or not. It is an extension of the first tool we developed which was responsible to collect data from the images and calculate the distributions and their deviations from the expected. We call the latter tool *compBennie*. The next section discusses the results we obtained when we tested the new steganalytic method using the UCID and also using a new dataset created by a smartphone.

4. Experiments and results

We evaluated the accuracy of our method in three stages. Firstly, we calculated the algorithm's false positive rate (FPR). Then we tested the validity of the method on the training set and finally on a set of images taken by a smartphone. Furthermore, we tested the efficiency of our

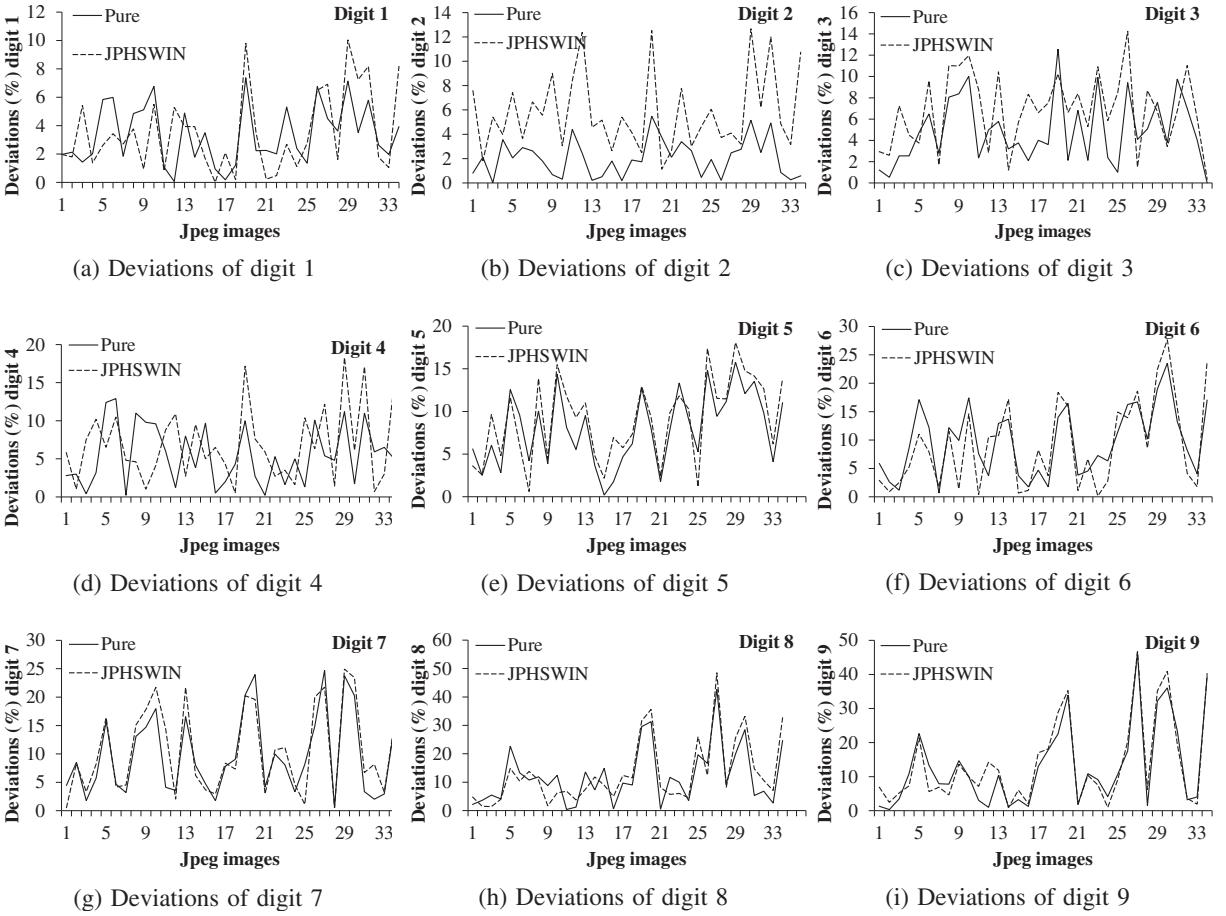

Fig. 1. Deviations of first digits for quality factor 75.

steganalytic tool (StegBennie) which utilizes the proposed method against a popular steganography detection tool (Stegdetect).

4.1. False positive rate

During the first phase we calculated the percentage of pure images that will be erroneously identified as suspicious (False Positive Rate – FPR). To achieve this goal we analysed all the pure images that were available to us (7 folders containing 1338 colour images each). Table 4 gathers the results from this procedure and also displays the time our tools needed to analyse each folder. Fig. 2 demonstrates the results of Table 4 in a chart. The vertical axis represents the % percentage of fault estimation for each quality factor. Images assessed as suspicious are presented in black colour. Apparently, about one image out of three or four will be considered as suspicious despite the fact that it will be clear.

4.2. Hit rates

The next step of the evaluation of the steganalytic ability of our method was to examine the percentage of malicious images successfully identified. For this task we used the

Table 3
Threshold for quality factors.

QF	Threshold
50	4.00
60	4.00
70	4.35
75	3.00
80	3.11
90	2.90

Table 4
The false positive rate (FPR) of our method.

QF	Suspicious	FPR	Processing time
50	444	33.18%	13 s
60	378	28.25%	18 s
70	243	18.16%	19 s
75	398	29.75%	20 s
80	323	24.14%	21 s
90	473	35.35%	23 s

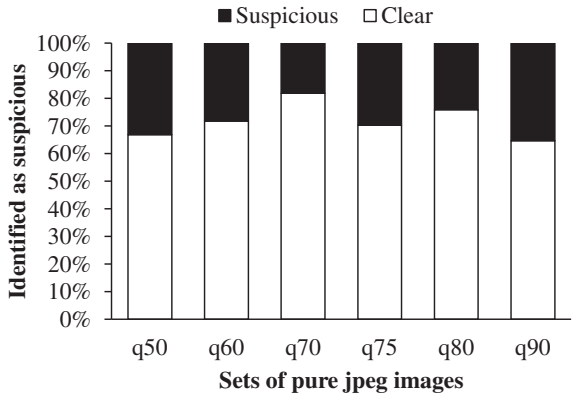

Fig. 2. False positive rate for pure images.

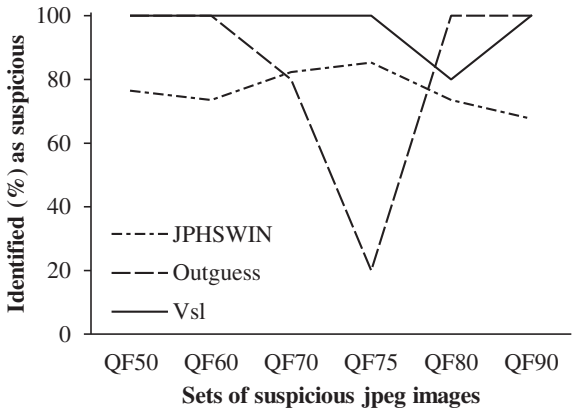

Fig. 3. Comparison of the hit rates for JPHSWIN, Outguess and Vsl.

images that are referred in Section 3. We had six folders with 204 randomly picked pure images and six folders with the same images containing hidden data, embedded with JPHSWIN, Outguess or Vsl. Table 5 aggregates the findings of this experiment. The second column of Table 5 demonstrates the false positive rate and the other columns show the hit rate of the method. In other words, columns 3, 4 and 5 illustrate the percentage of suspicious images flagged as malicious. Also, at Fig. 3 we provide a graphical depiction of the contents of Table 5. The vertical axis shows the % percentage of recognition of stego carriers. Lastly, Table 6 presents the number of images (%) that were identified as suspicious and declared as maliciously altered by either Outguess or Vsl.

In the final stage of our experiments we used a smartphone (HTC Desire). The device uses the same quantization tables we used previously in our research, the standardized IJG tables. The quality compressions of its camera are three; 'Fine' stands for QF = 90, 'High' stands for QF = 80 and 'Normal' for QF = 70. It also provides the opportunity to the user to decide about the resolution and the format ('widescreen' or 'standard') of the image. For this experiment we used about 150 images of different resolutions. The 'small' resolution was similar to the resolution that the UCID images had. The characteristics of the images we tested that had different resolutions can be seen at Table 7. Note that we also tested the accuracy of the method for a set of JPEG images with a different format than the standard ('widescreen').

4.3. Tests with real data

Here we used the same approach as described in the previous steps. Firstly, we measured the false positive rate.

Table 5
Hit rates.

QF	FPR	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 6
Effectiveness of algorithm identification.

QF	Identified Outguess	Identified Vsl
50	100	100
60	100	100
70	0	100
75	0	60
80	0	60
90	0	100

Table 8 illustrates the results and Fig. 4 demonstrates the findings graphically. The vertical axis represents the percentage of false positive results.

After acquiring good results for the false positive rate of the method, we embedded data in the images we took with the smartphone. We used again the same algorithms for this task; JPHSWIN, Outguess and Vsl. The results that came from this experiment are displayed at Table 9. Fig. 5 concatenates the results of Table 9. Again, the vertical axis stands for the percentage of successful recognition of stego carriers. Table 10 shows in which cases the method we used was able to identify the embedding algorithms.

Lastly, we used the same image sets to compare the speed of our steganalytic tool and its accuracy in identifying suspicious images against the popular JPEG steganalyzer Stegdetect.[2] We analysed 151 smartphone pure photos with both algorithms and the results of their performance can be seen in Table 11 (time is calculated in seconds).

For the needs of the second experiment we used the photos described in Table 9 (subset of the aforementioned 151). After grouping the 'small' and '1Mp' photos in nine sets of stego carriers, we ran the two tools to evaluate their steganalytic ability. The 'small' and '1Mp' photo sets were chosen because they have the same size properties as the images in the training set. The results from the latter experiment are concatenated in Table 12 and presented in Fig. 6.

[2] http://www.outguess.org/detection.php.

Table 7
Different formats of real data.

Folders	Pixels
small	640 × 480
1Mp	1280 × 960
3Mp	2048 × 1536
5Mp	2592 × 1952
wide1Mp	1280 × 768

5. Discussion of results

5.1. The false positive rate results

Looking back at Subsection 4.1 we conclude by the demonstrated results of Table 4 and Fig. 2 that the false positive rate (FPR) of our method is acceptable. We believe that the current FPR is a satisfactory percentage that could reduce the workload of a forensic examiner who performs steganalysis to JPEG images. Furthermore, if the inspected images are compressed with a quality factor of 70, the fault rate of the method is lower than 20%. The last column of Table 4 illustrates the approximate time in seconds that our steganalytic tool StegBennie needed in order to analyse the whole folder that was under examination. Each folder contained of 1338 JPEG images at a 512 × 318 resolution. The steganalytic tool is fast regardless of the compression's quality factor. As a consequence we proved that a combination of the method and a fast program like StegBennie can perform a trustworthy steganalysis to a folder containing JPEG images in less than half a minute.

5.2. Analysis of successful detection on the UCID set

After the embedding of text and doc files in various images, we presented Table 5 in Subsection 4.2. The conclusions that arise by examining this table and Fig. 3 are quite satisfactory. It seems that the ability of our methods to detect a suspicious JPEG image could be characterized as fairly strong. Moreover, the fault rate of our tool does not exceed the limits we saw at Table 4. The hit rates for JPHSWIN confirm that about 4 out of 5 malicious images will be successfully detected. At this point we should state that the results for Outguess and Vsl come from the examination of a small sample of images. The unsatisfactory hit rate for images which are compressed with QF = 75 comes from the fact that Outguess always uses this specific quality factor during the embedding of data into JPEG images.

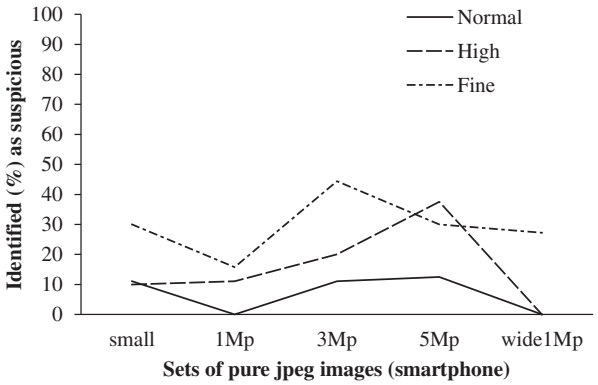

Fig. 4. Comparison of false positive rate among the different quality factors.

However, in Table 6 we can see that the number of images (%) that were identified as suspicious and declared as maliciously altered by either Outguess or Vsl is at a very good level considering images that were altered using QF = 50 or 60. Also, it seems that in most cases we are able to identify algorithms such as Vsl. From this table we deduce that for images with QF > 70 the identification of Outguess is unlikely to happen. As an overview the steganalytic ability of the method we introduce on the pseudo training image set is at a very good level considering the graphs and the tables that were presented.

5.3. Results for the smartphone images

At the final phase of our tests we examined a set of images captured by a smartphone (Subsection 4.3). By taking a closer look at Table 8 we can reach the conclusion that, generally, our initial assumptions that the method we introduce will successfully detect more than two

Table 8
FPR on real data.

QF	Resolution	Examined images	FPR (%)
Normal QF = 70	small	9	11.11
	1Mp	10	0
	3Mp	9	11.11
	5Mp	8	12.5
	wide1Mp	9	0
High QF = 80	small	10	10.0
	1Mp	9	11.11
	3Mp	10	20.0
	5Mp	8	37.5
	wide1Mp	10	0
Fine QF = 90	small	10	30
	1Mp	19	15.79
	3Mp	9	44.44
	5Mp	10	30.0
	wide1Mp	11	27.27

Table 9
Hit rates for real data.

QF	Resolution	JPHSWIN	Outguess	Vsl
Normal QF = 70	small	88.89	77.78	100.0
	1Mp	90.0	75.0	100.0
	3Mp	55.55	75.0	100.0
	5Mp	33.33	87.5	100.0
	wide1Mp	66.67	50.0	100.0
High QF = 80	small	100.0	100.0	100.0
	1Mp	66.67	100.0	100.0
	3Mp	50.0	100.0	100.0
	5Mp	50.0	71.43	100.0
	wide1Mp	60.0	100.0	100.0
Fine QF = 90	small	100.0	100.0	100.0
	1Mp	66.67	90.0	100.0
	3Mp	55.55	87.5	100.0
	5Mp	40.0	100.0	100.0
	wide1Mp	72.73	90.0	100.0

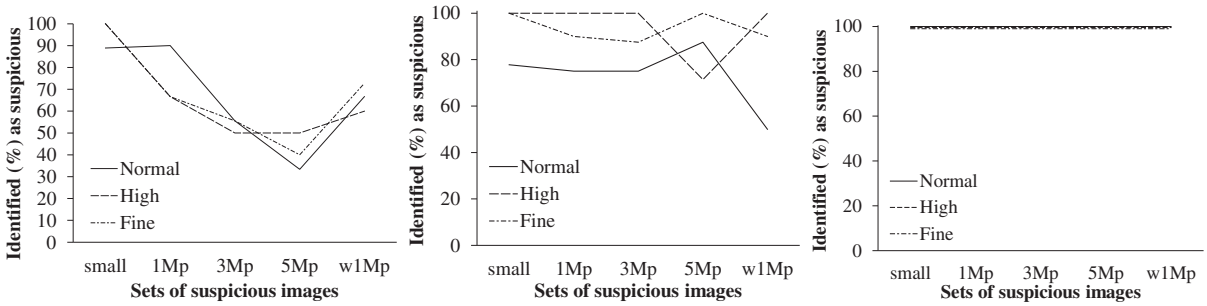

Fig. 5. Success of detection of a stego carrier.

thirds of clear images are correct. When examining images compressed with normal and high quality, the false positive rate was very low. Furthermore, there are cases that the accuracy of the method was exceptional (e.g. at normal quality when we examined images with resolution of 1Mp, the false positive rate was 0% either at standard format or at widescreen format). However, the false positive rate becomes larger when the quality factor of compression is 90. Results also demonstrate that the impact of the size of the image cannot affect dramatically the validity of the method. More experiments must be conducted to form a clear picture about the false positive rate of the method for images with a different format than the standard (e.g. 'widescreen'). An initial attempt to examine the disturbances on 1Mp images shows that the outcome will not be different if the image has a widescreen format. Fig. 4 visualizes the findings of Table 8. Our tool can detect quite precisely clear images which are compressed with normal quality (the false positive rate does not exceed 12.5%). The efficiency of our methods for the set of images in fine quality is not at very good standards compared to normal or high quality. However, in small sizes the percentage of the fault detection is fairly satisfactory.

The hit rates presented in Table 9 and Fig. 5 demonstrate the algorithm's correctness and efficiency. The first inference is the accuracy of detecting a stego carrier which was made by Vsl. This algorithm (Vsl) always causes critical deviations of digit 2, making detection by our tool a trivial task. StegBennie was able to correctly identify the use of Vsl in almost any case (Table 10).

We can also conclude that our method can adequately identify suspicious images up to 1Mp. For larger images containing secret data embedded by JPSWIN, the probability of correct detection drops. However, we were able to identify larger malicious images with a good hit rate when these images were manipulated by Outguess or Vsl. Recall that, in order to embed secret information, these two algorithms change the quantization tables of the original image. Thus, high detection rates are also very likely to be achieved if we examine stego carriers created by other algorithms that use the same technique (alteration of quantization tables).

The last remark has to do with the resolution of the images. Fig. 5a shows that hit rates decrease when the resolution of the image rises. However, if the format of an image changes (at the current experiment the standard format became widescreen format), the differences of the detection ability do not change dramatically. More experiments should be conducted to prove the accuracy of this assumption.

Table 11 validates the fact that our steganalytic tool which uses the proposed method, is twice faster than Stegdetect. In this phase we tested the tools with pure images produced by a smartphone. StegBennie's false positive rate is larger than Stegdetect's but we can still advocate for the capability of the method to efficiently distinguish 4 out of 5 pure images and thus reduce the investigator's workload. Table 12 confirms the fact that we can perform faster analysis of a folder which contains JPEG photos using StegBennie. Moreover, Fig. 6 depicts that our steganalytic tool was more accurate than Stegdetect when performed steganalysis on the same sets of stego carriers. Stegdetect was not able to identify Vsl because the latter steganographic tool (Vsl) is younger. Furthermore, Stegdetect did not manage to identify any stego carrier compressed with QF80 or QF90 despite the fact that it tracked about 30% of stegos compressed with QF70. On the other hand our steganalytic tool was able to accurately identify stego carriers produced by Outguess and Vsl.

To conclude, we illustrated that our method proved to be reliable when it was tested with real data. The results we

Table 10
Identification (%) of embedding algorithm.

	QF70					QF80					QF90				
	s	1Mp	3Mp	5Mp	w	s	1Mp	3Mp	5Mp	w	s	1Mp	3Mp	5Mp	w
Outguess	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
Vsl	100	100	89	100	89	100	89	88	90	63	100	100	100	100	100

Table 11
Comparison of steganalysis elapsed time and false positive rates between StegBennie and Stegdetect (smartphone photos).

Performance	StegBennie	Stegdetect
Time	19.037 s	38.222 s
FPR	17.22%	7.95%

Table 12
Steganography detection elapsed time for StegBennie and Stegdetect. Each cell represents the total time taken to process the entire category.

Stego carriers	StegBennie	Stegdetect
JPH70 (19 photos)	0.892 s	1.712 s
JPH80 (19 photos)	0.868 s	1.48 s
JPH90 (19 photos)	0.912 s	1.192 s
OUT70 (17 photos)	0.724 s	1.392 s
OUT80 (19 photos)	0.852 s	1.7 s
OUT90 (20 photos)	0.908 s	1.756 s
VSL70 (19 photos)	1.108 s	2.12 s
VSL80 (19 photos)	1.128 s	2.156 s
VSL90 (21 photos)	1.188 s	2.316 s

gathered are similar to those we saw at the training step in Section 4 and the comparison between our tool and the popular Stegdetect showed that StegBennie is faster and achieves better results.

6. Other steganalytic tasks

In this section we will discuss the results occurred by testing the ability of our methods to perform various steganalytic tasks. Not only are forensic examiners interested in detecting images with hidden data, but they also need to know if an image has been double compressed, cropped, blurred and generally if the image they examine has been modified. In order to meet the needs of this experiment we used the image set presented in Subsection 4.3. These are JPEG images captured by a smartphone. We used the 'small' and '1Mp' folders which contained images compressed with quality factor 70 and 80. We made this choice because of the low false positive rate of these images (Table 8). We used the open source image processing tool GIMP 2.0 for Windows to JPEG compress the images for a second time. GIMP also helped us crop, blur and change their settings and colours. Finally, we JPEG compressed a small set of BMP and PNG images which were taken from the internet, to see if this procedure can reveal the compression history of an image. The results of this experiment can be seen at Table 13.

Table 13
Detection rates for modified JPEG images.

Process		Images	Detection rate
Double JPEG	QF70 → QF70	20	5%
	QF70 → QF90	20	100%
	QF80 → QF80	20	10%
	QF80 → QF90	20	100%
Crop images		10	10%
Modify images		10	20%
JPEG compress PNG, BMP images		12	BMP: 83.3% PNG: 33.3%

Table 13 shows that it was not possible to detect the double-compression of a JPEG image when both compression passes used the same quality factor. However, we can flag as suspicious every JPEG image that was double compressed with different quality factors. On the contrary, our methods fail to recognize images that were cropped or modified by a program like GIMP. We must underline that for the current task we compressed the modified images with the initial quality factor. For example, after sharpening an image which had been initially compressed with quality factor 70, we saved the new modified image with the same quality factor (QF = 70). Finally, StegBennie is very likely to detect the process of compression of a BMP to a JPEG. However, the detection rate is lower for PNG images.

At any case, the most useful conclusion extracted by the final experiment is that the method is able to identify a double JPEG compressed image, when the quality factor used for the second compression is not the same as the initial one.

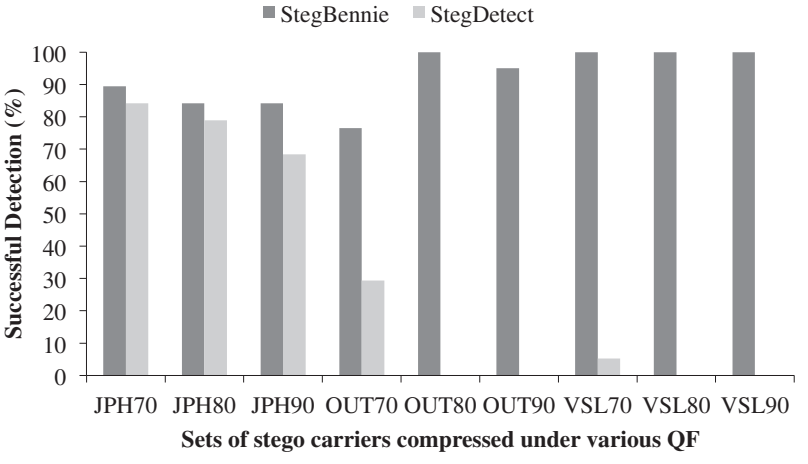

Fig. 6. Steganography detection ability of StegBennie compared to Stegdetect.

7. Conclusions and future work

In this paper we extended and further validated previous work of Fu et al. (2007) and applied it in the case of colour JPEG images. We used this extension to develop a new blind steganalytic method which utilizes Benford's Law introducing a new approach for the statistical steganalysis of JPEG images. Thus, the outcome of our work is dual: firstly, we studied the behaviour of the quantized DCT coefficients of colour images and confirmed the applicability of gBL; secondly we introduced a novel method which can assist with various steganalytic tasks and developed two fast and accurate new tools implementing it. By using those tools, a forensic expert can reveal the fingerprints left on compressed images by Outguess and Vsl. Results extracted by examining a large image set demonstrate the method's validity and its ability to detect suspicious images. False positive rates are fair and hit rates are satisfactory. Lastly, the tools can identify the use of Vsl in most of the cases.

A careful study of the deviations of the distributions of digit 2 revealed a stable, well defined and reliable behaviour compared to the disturbances that various steganographic techniques caused to the distributions of the other first digits of the quantized DCT coefficients. However, an improvement of the decision making module of our method could provide even better results than those we have already seen. The potential of mathematical algorithms that take into consideration the overall disturbances of all digits should be explored. The mathematical equation should weigh the impact of each digit based on information from previous investigation. In other words, machine learning techniques and neural network theory could be very helpful to optimize decision making. Also, research should be done to compare the efficiency of the method against modern well known state of the art techniques.

We would also like to study the consequences that the size of an image has to the distributions of the first digits in order to provide more accurate parameters, based on the quality factor, size and resolution of the colour JPEG image. In Subsection 5.3 we saw that the format of an image (standard or widescreen) does not have a significant impact on the outcome of our method. On the other hand, the amount of pixels is an important factor and influences the validity of the results.

In addition, we have to consider the effect of the size of the embedded data and measure its impact on the overall validity of the method. We conducted some experiments with a restricted quantity of images (taken from the 'small' folder) to measure the effect of the embedded message's size to the steganalysis process. The size and resolution of those images are almost equal to the characteristics of the training set. Unfortunately, the embedding capacity of those images is very small and we cannot reach a safe conclusion about the impact of the message's payload size to our method. However, it seems that when we are trying to detect stego carriers created by Outguess or Vsl, the size of the hidden message does not affect critically the steganalytic ability of the method and this finding can be explained by the fact that the aforementioned algorithms change image quantization tables. In the future we will have to test the steganalytic algorithm with larger images that will give us the chance to embed messages with various sizes and evaluate how they affect its validity.

Thus, further investigation and more thorough experiments must be conducted in order to form a suitable model which considers the quality factor, the number of components, the resolution of the JPEG image and the size of the embedded data.

In addition to the gBL, Fu et al. (2007) have also investigated the distributions of the first digits of the coefficients of the blocks of the JPEG images before the quantization step (during the compression of the image). They demonstrate that these distributions adhere to the original Benford's Law quite well. We have not yet applied this observation to our steganalytic method. Future development should consider this fact because it will probably provide the opportunity to ascertain the deviations of the distributions of the first digits of the block coefficients (before quantization) and the deviations of distributions of the quantized DCT coefficients of the JPEG image.

An encouraging conclusion about the application of our blind steganalytic method on different contexts is the success rate we achieved when we tried to identify images that were JPEG compressed for a second time. Table 13 showed that the method fails to detect an image which has been double compressed using the same compression quality factor. On the contrary, if the JPEG image has been double compressed and the second quality factor is different from the first, then our technique is 100% accurate. This finding might allow us to estimate the compression history of a double compressed JPEG image. In other words, we could use the deviations that our tools provide to estimate the initial quality factor of a JPEG image. We can then simply compare the set of distributions that an image provides with the sets of the expected distributions of all the known quality factors. If we find a set which contains distributions whose deviations of all digits between the expected and the gathered are minor, then we can draw conclusions about the initial quality factor of the image.

Acknowledgement

This work has been supported 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.

References

Benford F. The law of anomalous numbers. In: Proceedings of the American philosophical society; 1938. p. 551–72.

Chandramouli R, Kharrazi M, Memon N. Image steganography and steganalysis concepts and practice. In: Kalker T, editor. 2nd International Workshop on Digital Watermarking (IWDW 2003). In: Cox I, Ro Y, editors. Lecture Notes in Computer Science, Vol. 2939; 2004. p. 35–49.

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.

Fridrich J, Goljanb M, Du R. Steganalysis based on JPEG compatibility. In: Proc. conference on multimedia systems and applications IV. Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE); 2001. p. 275–80.

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

Jolion J. Images and Benford's law. Journal of Mathematical Imaging and Vision 2001;14(1):73–81.

Kodovsky J, Fridrich J, Holub V. Ensemble classifiers for steganalysis of digital media. IEEE Transactions on Information Forensics and Security 2012;7(2):432–44.

Lee K, Westfeld A, Lee. Category attack for LSB steganalysis of JPEG images. In: Digital watermarking (5th international workshop) IWDW 2006 Jeju Island, Korea, November 8–10, 2006. LNCS, Vol. 4283. Springer-Verlag; 2006. p. 35–48. Revised Papers.

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

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

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.

Schaefer G, Stich M. UCID – an uncompressed colour image database. In: Yeung M, editor. Proc. conference on storage and retrieval methods and applications for multimedia. In: Lienhart R, Li C, editors. Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), Vol. 5307; 2004. p. 472–80.

Schäfer C, Schräpler JP, Müller KR, Wagner GG. Automatic identification of faked and fraudulent interviews in surveys by two different methods. DIW-Diskussionspapiere 441. Deutsches Institut für Wirtschaftsforschung (DIW); 2004.

Solanki K, Sarkar A, Manjunath BS. YASS: yet another steganographic scheme that resists blind steganalysis. In: Furon T, editor. Information hiding. Lecture Notes in Computer Science, Vol. 4567; 2007. p. 16–31.

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

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

Zaharis A, Martini A, Tryfonas T, Illioudis C, Pangalos G. Lightweight steganalysis based on image reconstruction and lead digit distribution analysis. International Journal of Digital Crime and Forensics 2011;3(4):29–41.

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