# Attacking the OutGuess

Jessica Fridrich
Dept. of ECE
SUNY Binghamton
Binghamton, NY 13902-6000
001-607-777-2577
fridrich@binghamton.edu

Miroslav Goljan
Dept. of ECE
SUNY Binghamton
Binghamton, NY 13902-6000
001-607-777-5793
mgoljan@binghamton.edu

Dorin Hogea
Dept. of Computer Science
SUNY Binghamton
Binghamton, NY 13902-6000
001-607-777-5689
dhogea1@binghamton.edu

## ABSTRACT

In this paper, we describe new methodology for developing steganalytic methods for JPEG images. The proposed framework can be applied to virtually all current methods for JPEGs including OutGuess, F5, and J-Steg. It also enables accurate estimation of the length of the embedded secret message. The methodology is demonstrated on OutGuess 0.2.

## Categories and Subject Descriptors

Multimedia processing and coding, including multimedia content, analysis, content-based multimedia retrieval, multimedia security, audio/image/video processing, and compression

## General Terms

Algorithms, Design, Performance, Security

## Keywords

Steganography, steganalysis, JPEG, attack, OutGuess, F5

## 1. INTRODUCTION

The JPEG format is currently the most common format for storing image data. It is also supported by virtually all software applications that allow viewing and working with digital images. Recently, several steganographic techniques for data hiding in JPEGs have been developed: J-Steg [1], JP Hide&Seek [1], F5 [2], and OutGuess [3]. In all programs, message bits are embedded by manipulating the quantized DCT coefficients. J-Steg and OutGuess embed message bits into the LSBs of quantized DCT coefficients.

J-Steg with sequential message embedding is detectable using the chi-square attack [4]. J-Steg with random straddling as well as JP Hide&Seek are detectable using the generalized chi-square attack [5,6]. The chi-square attacks are not effective for F5 (F5 does not flip LSBs but decrements coefficient values by 1 if necessary) and for OutGuess (OutGuess preserves first-order statistics). The universal blind detectors pioneered by Farid [7] seem to be able to detect virtually every steganographic method after appropriate training on a database of stego and cover images, but the blind detectors do not allow accurate estimation of the embedded messages and it is not clear how their performance will scale to more diverse databases. A successful attack on the F5 algorithm has been recently reported in [8]. One important advantage of this approach is that one can obtain an accurate estimate for the length of the embedded secret message.

In the next section, we formulate a general methodology for developing steganalytic methods for JPEGs. We demonstrate the concepts by presenting a detection method for OutGuess in Section 3. The paper is concluded in Section 4, where we briefly describe how the same methodology can be used for detection of other programs, such as the F5 and J-Steg.

## 2. GENERAL METHODOLOGY

For most steganographic techniques, it is usually relatively easy to identify a macroscopic quantity $S(p)$ that predictably changes (for example, monotonically increases) with the length of the embedded secret message $p$. Let us assume that the functional form of $S$ is known or can be guessed from experiments. The function $S$ may depend on several undetermined parameters. We can attempt to determine those parameters by estimating some extreme values of $S$, such as $S(0)$ ($S$ for the cover image) or $S(p_{max})$ (for the stego image with maximal message). Once the parameters have been determined, one can calculate an estimate of the unknown message length $q$ by solving the equation $S(q) = S_q$ for $q$, where $S_q$ is the value of $S$ for the stego image under investigation. An important advantage of this approach is that the detection is *threshold-free* and an estimate for the length of the secret message can be obtained.

In this paper, we show what macroscopic quantities are useful for detection and how to obtain an estimate of the cover image steganographic methods that embed message bits in quantized JPEG DCT coefficients. We crop the (decompressed) stego image by 4 pixels and recompress it using the quantization table of the stego image. Because of the cropping, the newly calculated DCT coefficients will not exhibit clusters due to quantization. Also, because the cropped stego image is visually similar to the cover image, macroscopic characteristics, such as $S$, will be approximately preserved. This JPEG image will then be used to determine the parameters in the functional form of $S$.

In the past, we have successfully applied this approach to the F5 algorithm. Because the F5 modifies the histogram of DCT coefficients in a predictable manner, we chose the individual

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Conference '00*, Month 1-2, 2000, City, State.

Copyright 2000 ACM 1-58113-000-0/00/0000…$5.00.

histograms of DCT coefficients as the macroscopic quantity *S*. Details of this approach can be found in [8].

Because OutGuess preserves the first order statistics (histogram), we cannot use the same approach. Instead, we turned our attention to the measure of discontinuities along the boundaries of 8×8 pixel blocks. Also, we utilize the fact that the embedding process and the correction step are simple LSB flipping operations.

## 3. BREAKING OUTGUESS

The OutGuess steganographic algorithm was proposed by Neils Provos [3] to counter the statistical chi-square attack [4]. In the first pass, similar to J-Steg, OutGuess embeds message bits along a random walk into the LSBs of coefficients while skipping 0's and 1's. After embedding, the image is processed again using a second pass. This time, corrections are made to the coefficients to make the stego image histogram match the cover image histogram. Because the chi-square attack is based on analyzing first-order statistics of the stego image, it cannot detect messages embedded using OutGuess. Provos also reports that the corrections are made in such a manner to avoid detection using his generalized chi-square attack [5].

In our attack on OutGuess, we use the fact that the embedding mechanism in OutGuess is overwriting the LSBs. This means that embedding another message into the stego image will partially cancel out and will thus have a different effect on the stego image than on the cover image.

In the rest of this text, we will work with grayscale images. Extension to color images should be obvious. Let $h_d$, $d = \ldots, -2, -1, 0, 1, 2, \ldots$ be the histogram of the quantized DCT coefficients from the cover image. Let $P$ be the total number of coefficients different from 0 and 1:

$$P = \sum_{\substack{i\neq 0\\ i\neq 1}} h_i \,.$$

We will call those coefficients usable coefficients. OutGuess first calculates the maximal length of a randomly-spread message that can be embedded in the image while making sure that one will be able to make corrections to adjust the histogram to its original values. After embedding $m$ pseudo-random bits in the LSBs of the cover-image in randomly selected usable coefficients, the histogram values $(h_{2i}, h_{2i+1})$ will be changed to

$$h_{2i} \rightarrow h_{2i} - \alpha(h_{2i} - h_{2i+1}),$$
$$h_{2i+1} \rightarrow h_{2i+1} + \alpha(h_{2i} - h_{2i+1}),$$

where $2\alpha = m/P$. Let us assume that, for example, $h_{2i} > h_{2i+1}$. After embedding, there must be enough coefficients with value $2i+1$ to make necessary corrections. Thus, $h_{2i+1} - 2\alpha\, h_{2i+1} = \alpha\,(h_{2i} - h_{2i+1})$, which gives

$$\alpha_i = \frac{h_{2i+1}}{h_{2i+1} + h_{2i}} \,.$$

This condition must be satisfied for all histogram pairs $(h_{2i}, h_{2i+1})$. Thus, the maximal message size that can be embedded in the image with appropriate corrections is $2aP$, where $a = \min_i \alpha_i$.

After embedding a message of size $2paP$ bits, $0\leq p\leq 1$, in the cover image (we call such a message a $p$-percent message), due to the correction step, the number of changes for values $2i$ and $2i+1$ are both $pah_{2i}$, assuming $h_{2i} > h_{2i+1}$. Thus, the total number of changes (both due to embedding and correction) is

$$T_p = 2pa\sum_{i\neq 0}\overline{h}_{2i} = paP + pa\sum_{i\neq 0}\left|\overline{h}_{2i} - \underline{h}_{2i}\right|, \tag{1}$$

where $\overline{h}_{2i} = \max(h_{2i}, h_{2i+1})$ and $\underline{h}_{2i+1} = \min(h_{2i}, h_{2i+1})$ for each $i$. The first term is due to message embedding, the second term due to corrections.

Because OutGuess introduces random changes into the quantized coefficients, the spatial discontinuities at the boundaries of all 8×8 blocks will increase. We will measure the discontinuity using the blockiness measure (3). For detection, we will inspect the increase of this blockiness measure after embedding a 100% message again using OutGuess. This increase will be smaller for the stego image than for the cover image because of the partial cancellation of changes. This difference will form the basis of our message length estimation.

To mathematically analyze the proposed idea, we first calculate the number of changes after consecutive embedding of two messages in one image. Given a set of $n$ integers, if we randomly select a subset $S$ consisting of $s$ integers and flip their LSBs and then do the same again with another randomly chosen subset $R$ with $r$ integers, the number of integers with flipped LSBs will be equal to $|S \div R|$, where the symbol "$\div$" denotes the symmetric set difference and $|A|$ is the cardinality of $A$. This is because the integers in $S\cap R$ will be flipped twice and thus unchanged. Consequently, the total expected number of integers with flipped LSBs will be $r + s - 2rs/n$.

Therefore, if we embed an additional message of size $2qaP$, $0\leq q\leq 1$, into the image that already holds $2paP$ bits, the expected values of changes for the values $2i$ and $2i+1$ are

$pa\,\overline{h}_{2i} + qa\,\overline{h}_{2i} - 2pqa^2\,\overline{h}_{2i} = a\,\overline{h}_{2i}\,(p + q - 2pqa)$ and

$pa\,\overline{h}_{2i} + qa\,\overline{h}_{2i} - 2pqa^2\,\overline{h}_{2i}^{\,2}/\underline{h}_{2i+1} = a\,\overline{h}_{2i}\,(p + q - 2pqa\,\overline{h}_{2i}/\underline{h}_{2i+1})$ ,

respectively. Thus, the total number of expected changes in the cover image after consecutive embedding of two independent randomly-spread messages of size $2paP$ and $2qaP$ bits, $0 \leq p, q \leq 1$, is

$$T_{pq} = 2a\sum_{i\neq 0}\overline{h}_{2i}\left(p + q - apq\left(1 + \frac{\overline{h}_{2i}}{\underline{h}_{2i}}\right)\right). \tag{2}$$

The measure of blockiness at the block boundaries will be calculated using the following formula

$$\begin{aligned} B = {} & \sum_{i=1}^{\lfloor M-1/8\rfloor}\sum_{j=1}^{N}\left|g_{8i,j} - g_{8i+1,j}\right| + \\ & + \sum_{j=1}^{\lfloor N-1/8\rfloor}\sum_{i=1}^{M}\left|g_{i,8j} - g_{i,8j+1}\right| \end{aligned} \tag{3}$$

where $g_{ij}$ are pixel values in an $M\times N$ grayscale image and $\lfloor x\rfloor$ denotes the integer part of $x$.

We have a compelling experimental evidence that the blockiness $B$ increases linearly with the number of DCT coefficients with flipped LSBs. The slope of this linear dependency is largest for the cover image and becomes smaller for an image that already contains a message. We use this slope as the macroscopic quantity $S$ to estimate the message length.

The detection will consist of the following steps:

1. Decompress the stego image, calculate its blockiness and denote *Bs*(0).
2. Using OutGuess, embed the maximal length message in the stego image (2*aP* bits), decompress, calculate the blockiness, and denote *Bs*(1). Calculate the slope $S = Bs(1) - Bs(0)$.
3. Crop the decompressed stego image by 4 columns. This image will be the baseline image that we will use to calibrate the slope. Compress the baseline image using the same JPEG quantization matrix as in the stego image. Decompress to the spatial domain and calculate its blockiness *B*(0).
4. Using OutGuess, embed the maximal length message in the cropped image and calculate the blockiness *B*(1).
5. Use the embedded image from Step 4 and, again, using OutGuess, embed the maximal length message in it denoting its blockiness *B*1(1).
6. Calculate the secret message length using Equation (4) (see the derivation below)

The slope $S_0 = B(1) - B(0)$ is what we would expect for the original cover image ($p = 0$). The slope $S_1 = B1(1) - B(1)$ is what we would obtain for an image with maximal embedded message ($p = 1$). The slope $S = Bs(1) - Bs(0)$ for the stego image will be somewhere in between these two slopes, $S\in[S_1,S_0]$ corresponding to an unknown message length *p*. We use linear interpolation to obtain the formula for *p*, $S = S_0 - p(S_0 - S_1)$, which gives us

$$p = \frac{S_0 - S}{S_0 - S_1}\ . \qquad (4)$$

The linear interpolation and Equation (4) can be justified using Equation (2) for the number of changes. Because the blockiness is a linear function of the number of DCT coefficients with flipped LSBs, we can write $B(p) = c + dT_p$, where $T_p$ is the number of coefficients with flipped LSBs after embedding a message of length 2*paP* bits, and *c* and *d* are constants. Using (2) we can write

$$S_1 = B1(1) - B(1) = d(T_{11} - T_{10}) = 2ad\sum_{i\neq 0}\overline{h}_{2i}\left(1 - a\left(1 + \frac{\overline{h}_{2i}}{\underline{h}_{2i}}\right)\right)$$

$$S_0 = B(1) - B(0) = d(T_{10} - T_{00}) = 2ad\sum_{i\neq 0}\overline{h}_{2i}$$

$$S = Bs(1) - Bs(1) = d(T_{p1} - T_{p0}) = 2ad\sum_{i\neq 0}\overline{h}_{2i}\left(1 - ap\left(1 + \frac{\overline{h}_{2i}}{\underline{h}_{2i}}\right)\right)$$

which, after simple algebra, confirms Equation (4). Equation (4) generally provides an accurate estimate of the secret message length. However, there are some situations when a large error may occur. This happens when the image sent to OutGuess is already a JPEG file. OutGuess always decompresses the cover image to the spatial domain and then recompresses it using a specified quality factor. The message is then embedded into this recompressed image by modifying the LSBs of DCT coefficients. If the quality factor $Q_c$ of the cover image is different from the quality factor for the stego image $Q_s$, the stego image is double-compressed (double quantized) and can have very singular properties in the frequency domain, such as a "jagged" DCT histogram. The baseline image obtained by cropping and recompressing the stego image will have macroscopic characteristics that correspond to the cover image but not to the double-compressed image. This may cause a large error especially when $Q_c < Q_s$. An obvious remedy to this is to try to recognize the fact that the stego image has been double compressed and then estimate the original quality factor $Q_c$. Fortunately, OutGuess preserves the histogram and this helps us to recover $Q_c$. Thus, in the final version of the detection algorithm, Step 3 is replaced with:

3'. Crop the decompressed stego image by 4 columns. Compress the cropped image using $Q_c$, decompress, and recompress using $Q_s$ – a process that effectively simulates what happens during embedding. Decompress and calculate the blockiness *B*(0).

There are probably many ways how to estimate the quality factor $Q_c$ from the stego image. We opted for the following simple algorithm. Let $h_d\,(i, j)$ is the histogram of values of the (*i*, *j*)-th DCT mode for the stego image and let $h_d\,(i, j, Q)$ is the same for the cropped stego image that has been compressed using the quality factor *Q*, decompressed and recompressed using the stego image quality factor $Q_s$. We calculate $Q_c$ as the quality factor that minimizes the difference between $h_d\,(i, j, Q)$ and $h_d\,(i, j)$ for those DCT modes (*i*, *j*) that correspond to the lowest-frequency DCTs (1,2), (2,1), (2,2) (the mode (1,1) is the DC term):

$$Q_c = \arg\min_Q \sum_{(i,j)}\sum_d \left|h_d(i,j) - h_d(i,j,Q)\right|^2 .$$

We have tested this algorithm on 70 test grayscale 600×800 JPEG images with quality factors ranging from 70 to 90 and a fixed stego image quality factor $Q_s = 80$. In all but four cases we estimated the cover image quality factor correctly.

The same database of images was used for evaluation of the performance of our detection method. Among the 70 test images, 24 of them were processed using OutGuess with message sizes ranging from the maximal capacity to zero. Because the detection algorithm contains randomization, we have repeated the detection 10 times for each image and averaged the *p* values (4). The results are shown in Figure 1. On the *y* axis is the relative number of changes due to embedding $T_p/aP$ (see Equation (1)) and on the *x* axis is the image number. Assuming the distribution of the difference between the estimated and actual values is Gaussian, the estimation error is –0.0032 ± 0.0406. From our experiments with Equation (1) on test images, we determined that the number of changes due to the correction step is about 1/3 of the changes due to message embedding. Thus, on average the total number of changes due to embedding *m* bits is $T_p = m/2\,(1+1/3)$. Consequently, the error for the estimated message length *m* is –0.48 ± 6 % of total capacity.

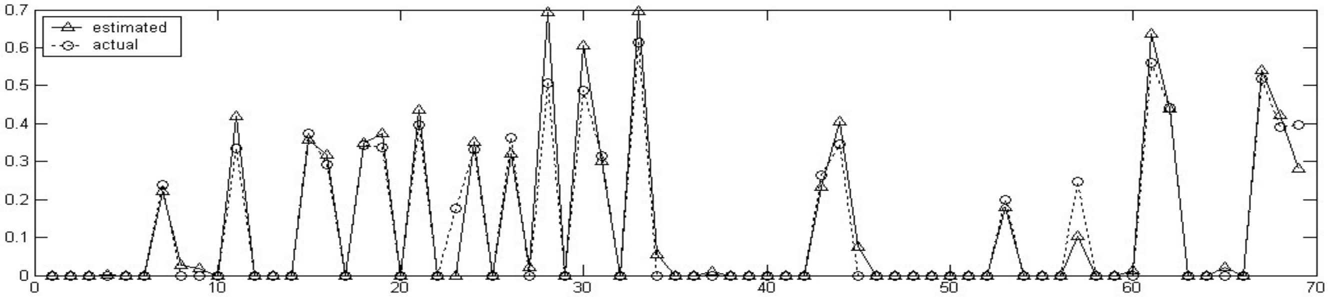

**Figure 1. The actual relative number of changes $T_p/aP$ (circles) compared to the calculated number of changes (triangles) for 70 test JPEG images resized to 600×800 pixels obtained using a digital camera Kodak DC 290**

## 4. CONCLUSION

In this paper, we describe a threshold-free detection methodology for attacking steganographic methods that embed data by modifying quantized DCT coefficients. The detection starts with identifying a macroscopic quantity $S(p)$ that predictably changes with the length of the embedded message. We show how to determine the parameters in $S$ by calculating $S(0)$ and $S(1)$ for an approximation to the cover image obtained by cropping the stego image and recompressing it. Using the values $S(0)$ and $S(1)$, it is possible to calculate an estimate of the length of the embedded message $p$. For OutGuess, we take the increase in spatial blockiness as a function of $p$ as the macroscopic quantity $S$. For the database of 70 grayscale images, the estimated relative number of modifications due to embedding is quite close to the actual numbers with the standard deviation for the error of 4% of the total image capacity.

The detection methodology is based on the assumption that the macroscopic quantity $S$ behaves approximately the same for the cover image and the cropped recompressed stego image. Although, this assumption has been verified experimentally, it deserves a more formal mathematical approach. It would be especially useful to automatically detect cases when this assumption is not satisfied and thus the result of the detection may be inaccurate.

For F5, we can take the individual histograms of low-frequency DCT coefficients as the quantity $S$ (for details, see [8]). For J-Steg (including the version of J-Steg with random straddling), one can also use the histogram because it changes predictably with the length of the embedded message.

One of the lessons learned from this paper is that in order to develop a high-capacity steganographic method for JPEGs, one needs to avoid making predictable changes to some macroscopic characteristics of the JPEG file. However, this task seems to be quite difficult if we insist on embedding one bit in each non-zero DCT coefficient. Also, another lesson is that one should abandon the concept of LSB flipping for embedding and instead use incrementing/decrementing the coefficient values as already pointed out in [2].

## 5. ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under a research grant number F30602-02-2-0093. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

## 6. REFERENCES

[1] Steganography software for Windows, http: //members.tripod.com/steganography/stego/ software.html

[2] Westfeld, A. High Capacity Despite Better Steganalysis (F5– A Steganographic Algorithm). In: Moskowitz, I.S. (eds.): Information Hiding. 4th International Workshop. Lecture Notes in Computer Science, Vol.2137. Springer-Verlag, Berlin Heidelberg New York, 2001, pp. 289–302

[3] Provos, N. Defending Against Statistical Steganalysis. Proc. 10th USENIX Security Symposium. Washington, DC, 2001

[4] Westfeld, A. and Pfitzmann, A. Attacks on Steganographic Systems. In: Pfitzmann A. (eds.): 3rd International Workshop. Lecture Notes in Computer Science, Vol.1768. Springer-Verlag, Berlin Heidelberg New York (2000), pp. 61–75

[5] Provos, N. and Honeyman, P. Detecting Steganographic Content on the Internet. CITI Technical Report 01-11, 2001

[6] Westfeld, A. Detecting Low Embedding Rates. 5th Information Hiding Workshop. Nooerdwijkerhout, Netherlands, Oct. 7–9, 2002

[7] Farid, H. and Siwei Lyu. Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines. 5th Information Hiding Workshop, Noordwijkerhout, Netherlands, Oct. 7–9, 2002

[8] Fridrich, J., Goljan, M., and D. Hogea. Steganalysis of JPEG Images: Breaking the F5 Algorithm. 5th Information Hiding Workshop, Noordwijkerhout, Netherlands, Oct. 7-9, 2002

[9] Fridrich, J., Goljan, M., and Hogea, D.: New Methodology for Breaking Steganographic Techniques for JPEGs. Submitted to SPIE: Electronic Imaging 2003, Security and Watermarking of Multimedia Contents. Santa Clara, California, 2003