Practical Steganalysis of Digital Images – State of the Art

Jessica Fridrich*, Miroslav Goljan
SUNY Binghamton, Department of Electrical Engineering, Binghamton, NY 13902-6000

ABSTRACT

Steganography is the art of hiding the very presence of communication by embedding secret messages into innocuous looking cover documents, such as digital images. Detection of steganography, estimation of message length, and its extraction belong to the field of steganalysis. Steganalysis has recently received a great deal of attention both from law enforcement and the media. In our paper, we classify and review current stego-detection algorithms that can be used to trace popular steganographic products. We recognize several qualitatively different approaches to practical steganalysis – visual detection, detection based on first order statistics (histogram analysis), dual statistics methods that use spatial correlations in images and higher-order statistics (RS steganalysis), universal blind detection schemes, and special cases, such as JPEG compatibility steganalysis. We also present some new results regarding our previously proposed detection of LSB embedding using sensitive dual statistics. The recent steganalytic methods indicate that the most common paradigm in image steganography – the bit-replacement or bit substitution – is inherently insecure with "safe capacities" far smaller than previously thought.

Keywords: Steganalysis, steganography, LSB embedding, attacks, detection, covert communication

1. INTRODUCTION

Steganography is the art of invisible communication. Its purpose is to hide the very presence of communication by embedding messages into innocuous-looking cover objects. In today's digital world, invisible ink and paper have been replaced by much more versatile and practical covers for hiding messages – digital documents, images, video, and audio files. As long as an electronic document contains perceptually irrelevant or redundant information, it can be used as a "cover" to hide secret messages. In this paper, we focus on issues related to detection of hidden information in digital images in the passive warden scenario.

Each steganographic communication system consists of an embedding algorithm and an extraction algorithm. To accommodate a secret message, the original image, also called the cover-image, is slightly modified by the embedding algorithm. As a result, the stego-image is obtained. The set of images from which cover-images are being drawn is part of the communication system.

As in cryptography, we should assume that the steganographic method is publicly known with the exception of a secret key. The method will be called secure if the stego-images do not contain any detectable artifacts due to message embedding. In other words, the set of stego-images should have the same statistical properties as the set of cover-images. If there exists an algorithm that can guess whether or not a given image contains a secret message with a success rate better than random guessing, the steganographic system is considered broken. Several definitions of steganographic security were proposed in the literature[2,4,15]. The problem with most definitions is that they assume observers with unlimited computational power and detailed statistical knowledge of the source of the cover-images. In practice, these assumptions are rarely satisfied or feasible to obtain. Attempts to define the concept of steganographic security in a relevant and practical way include the recent work of Katzenbeisser et al[15].

The choice of cover images is important and influences the security in a major way. Images with a low number of colors, computer art, images with a unique semantic content, such as fonts, should be avoided. Some steganographic experts

[*] fridrich@binghamton.edu; phone: 1 607 777-2577; fax: 1 607 777-2577; http://www.ssie.binghamton.edu/fridrich; SUNY Binghamton; T. J. Watson School of Engineering, Dept. of Electrical Engineering, Binghamton, NY USA 13902-6000

recommend grayscale images as the best cover-images[3]. They recommend uncompressed scans of photographs or images obtained with a digital camera containing a high number of colors and consider them safe for steganography.

Obviously, the less information we embed into the cover-image, the smaller the probability of introducing detectable artifacts by the embedding process. Each steganographic method seems to have an upper bound on the maximal safe message length (or the bit-rate expressed in bits per pixel or sample) that tells us how many bits can be safely embedded in a given image without introducing any statistically detectable artifacts. Determining this maximal safe bit-rate is a very difficult task even for the simplest methods. Surprisingly little has been published on this important topic. An exception is the work by Chandramouli et al.[5] who gave a theoretical analysis of the maximal safe bit-rate for the least significant bit (LSB) embedding in the spatial domain.

Dozens of steganographic techniques are today available on the Internet[22]. Most are creations of amateur enthusiasts available for free, while others are products of private companies and can be purchased for a small fee. Most of those techniques utilize the LSB embedding applied either directly to pixel values, to indices in palette images (EZ Stego[22]), or to quantized DCT coefficients for the JPEG format (J-Steg[22], JPHide&Seek[22], OutGuess[17]). While most of the publicly available techniques are very simplistic and relatively unsophisticated, reliable detection mechanisms are mostly not available. As any modern communication technology, steganography can be misused by criminals for planning and coordinating criminal activities. By embedding messages in images and posting them on binary newsgroups or public sites, it is difficult to discover the communication act itself and trace the recipient of the message.

Provos[18] carried out an extensive analysis of JPEG images downloaded from eBay. Using his steganalytic software he identified several thousands of “suspicious” images embedded with J-Steg and JP Hide&Seek. A dictionary attack was then applied in an attempt to recover the hidden message. Although this experiment did not reveal presence of any secret messages, it does not prove that criminals are not using steganography for communication.

Pfitzmann and Westfeld[20] introduced a method based on statistical analysis of Pairs of Values (PoVs) that are exchanged during message embedding (see Section 2.2 of this paper). Pairs of Values that differ in the LSB only, for example, could form these PoVs. This method provides very reliable results when we know the message placement (such as sequential). Provos[17] noted that the method could still be used for detection of randomly scattered messages by applying the same idea to smaller portions of the image. However, he gives no further details or estimates of false positives and negatives for this generalized approach.

Fridrich et al.[10,11] introduced a powerful steganalytic method (RS steganalysis) for detection of LSB embedding that utilizes sensitive dual statistics derived from spatial correlations in images (see Section 2.3 of this paper). In a typical cover-image, the LSB plane can be predicted to some degree from the remaining 7 bit-planes. This prediction becomes less reliable as the LSB is randomized. This can be captured mathematically and used for building a sensitive and accurate steganalytic method. For high quality images taken with a digital camera or a scanner, the RS steganalysis indicates that the safe bit-rate is less than 0.005 bits per sample.

Fridrich et al.[8] have recently shown that cover-images stored in the JPEG format are a very poor choice for steganographic methods that work in the spatial domain (see Section 2.4 of this paper). This is because the quantization introduced by JPEG compression can serve as a "semi-fragile watermark" or a unique fingerprint that can be used for detection of very small modifications of the cover image by inspecting the compatibility of the stego-image with the JPEG format. Indeed, changes as small as flipping the LSB of one pixel can be reliably detected. Consequently, one should avoid using decompressed JPEG images as covers for spatial steganographic methods, such as the LSB embedding or its variants.

Farid[6] developed a universal blind detection scheme that can be applied to any steganographic scheme after proper training on databases of original and cover-images (see Section 2.5 of this paper). He uses an optimal linear predictor for wavelet coefficients and calculates the first four moments of the distribution of the prediction error. The statistics is calculated for a large database of original and stego-images. Fisher linear discriminant statistical clustering is then used to find a threshold that separates stego-images from original images. Farid demonstrates the performance on J-Steg, both versions of OutGuess[17], and EZ Stego[22]. It appears that the selected statistics is rich enough to cover a very wide range of steganographic methods.

In the rest of the introduction, we mention other steganalytic methods previously proposed in the literature. Fridrich et al.[9] developed a steganographic method for detection of LSB embedding in 24-bit color images (the Raw Quick Pairs – RQP method). The RQP method is based on analyzing close pairs of colors created by LSB embedding. It works reasonably well as long as the number of unique colors in the cover image is less than 30% of the number of pixels. The RQP method can only provide a rough estimate of the size of the secret message. The results become progressively unreliable once the number of unique colors exceeds about 50 percent of the number of pixels. This frequently happens for high resolution raw scans and images taken with digital cameras stored in an uncompressed format. Another disadvantage of the RQP method is that it cannot be applied to grayscale images.

Johnson and Jajodia[13,14] pointed out that steganographic methods for palette images that preprocess the palette before embedding are very vulnerable. Some steganographic programs (e.g., S-Tools) create clusters of close palette colors that can be swapped for each other to embed message bits. These programs decrease the color depth and then expand it to 256 by making small perturbations to the colors. This preprocessing, however, will create suspicious pairs (or clusters) of colors that can be easily detected.

In this paper, we give a thorough review of practical steganalytic methods proposed in the literature. We also present some new results on dual statistics methods in Section 2.3 and on the JPEG compatibility method in Section 2.4. We close the paper by making an observation about security and safe message length of steganographic methods that utilize bit-replacement. In this paper, we will assume that the embedded message is a pseudo-random bit-stream, such as a bit-stream obtained by encryption or compression.

2. STEGANALYTIC METHODS

2.1. Visual attacks

Most steganographic programs embed message bits either sequentially or in some pseudo-random fashion. In most programs, the message bits are chosen non-adaptively independently of the image content. If the image contains connected areas of uniform color or areas with the color saturated at either 0 or 255, we can look for suspicious artifacts using simple visual inspection after preprocessing the stego-image. Even though the artifacts cannot be readily seen, we can plot one bit-plane (for example, the LSB plane) and inspect just the bit-plane itself. This attack is especially applicable to palette images for LSB embedding in indices to the palette. If, at the same time, the message is embedded sequentially, one can have a convincing argument for the presence of steganographic messages in an image. However, as Pfitzmann and Westfeld report[20], it may be impossible to distinguish noisy images or highly textured images from stego-images using this technique. Although visual attacks are simple, they are hard to automatize and their reliability is highly questionable.

2.2. Statistical analysis of pairs of values (histogram analysis)

Pfitzman and Westfeld[20] introduced a powerful statistical attack that can be applied to any steganographic technique in which a fixed set of Pairs of Values (PoVs) are flipped into each other to embed message bits. For example, the PoVs can be formed by pixel values, quantized DCT coefficients, or palette indices that differ in the LSB. Before embedding, in the cover image the two values from each pair are distributed unevenly. After message embedding, the occurrences of the values in each pair will have a tendency to become equal (this depends on the message length). Since swapping one value into another does not change the sum of occurrences of both colors in the image, one can use this fact to design a statistical Chi-square test. We can test for the statistical significance of the fact that the occurrences of both values in each pair are the same. If, in addition to that, the stego-technique embeds message bits sequentially into subsequent pixels/indices/coefficients starting in the upper left corner, one will observe an abrupt change in our statistical evidence as we encounter the end of the message.

For example, for a palette image, we have at most 256 colors c_i in the palette, which means at most 128 PoVs. For the i-th pair, $i = 1, \ldots, k$, we define $n_i' = 1/2$(number of indices in the set $\{c_{2i}, c_{2i+1}\}$) and n_i = number of indices equal to c_{2i} . n_i' is the theoretically expected frequency if a random message has been embedded, and n_i is the actual number of occurrences of color c_{2i}. We can now perform a Chi-square test for the equality of n_i' and n_i. The Chi-square statistics is calculated as

$\chi_{k-1}{}^2 = \sum_{i=1}^{k} \frac{(n_i - n'_i)^2}{n'_i}$ with k–1 degrees of freedom, the p-value $p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi_{k-1}{}^2} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1}\,dx$ expressing the probability that the distributions of n_i' and n_i are equal.

For a sequentially embedded message, one can scan the image in the same order in which the message has been embedded and evaluate the p value for the set of all already visited pixels. The p value will at first be close to 1 and then it suddenly drops to 0 when we arrive at the end of the message. It will stay at zero till we get to the lower right corner. Thus, this test enables us not only to determine with a very high probability that a message has been embedded, but also determine the size of the secret message.

If the message-carrying pixels in the image are selected randomly rather than sequentially, this test becomes less effective unless majority of pixels have been used for embedding (message size compared to the number of pixels in the image). Provos[17] noted that if this technique is applied to different smaller areas in the image, the p-value will fluctuate with a decreasing degree of fluctuation as the message length increases. This is because a randomly spread message will be due to chance more concentrated in some areas than in others. He claims that by quantifying this observation, one could in principle detect even randomly scattered messages and estimate their length. Unfortunately, Provos does not provide any further statistical analysis of this observation or any other details supporting this claim.

Finally we note that any steganalytic technique based on the analysis of sample counts (the histogram) will be easy to circumvent. Provos[17] shows how one can design a JPEG embedding technique (OutGuess 0.2) that will preserve the original counts of samples in their PoVs and thus avoid message detection using Pfitzmann and Westfeld's steganalysis.

2.3. Dual statistics methods

Statistical methods that start with sample counts, such as the methods by Westfeld[20] or Provos[17] neglect a large amount or very important information – the placement of pixels in the stego-image. It is intuitively clear that utilizing the spatial correlations in the stego-image, one should be able to build much more reliable and accurate detection. However, it is not easy to uncover and quantify the weak relationship between some pseudo-random components present in the image (e.g., the LSB plane) and the image itself. Once this relationship is quantified using a measure, one could study how this measure changes with message embedding. The derived relationship can serve as a basis for steganalytic techniques. Below, we show how the ideas presented above can be realized for LSB embedding in the spatial domain.

Let us assume that we have a cover image with $M\times N$ pixels and with pixel values from a set P. For example, for an 8-bit grayscale image, $P = \{0, \ldots, 255\}$. We will capture the spatial correlations using a discrimination function f that assigns a real number $f(x_1, \ldots, x_n)\in\mathbf{R}$ to a group of pixels $G = (x_1, \ldots, x_n)$. In this paper, we use the function f defined as

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| , \tag{1}$$

which measures the smoothness of G – the noisier the group G is, the larger the value of the discrimination function becomes.

The LSB embedding increases the noisiness in the image, and thus we expect the value of f to increase after LSB embedding. The LSB embedding process can be conveniently described using a flipping function F_1: $0 \leftrightarrow 1$, $2 \leftrightarrow 3$, …, $254 \leftrightarrow 255$. Changing the LSB of the gray level x is the same as applying flipping F to x. We also define a dual concept called *shifted* LSB flipping F_{-1} as $-1 \leftrightarrow 0$, $1 \leftrightarrow 2$, $3 \leftrightarrow 4$, …, $253 \leftrightarrow 254$, $255 \leftrightarrow 256$, or

$$F_{-1}(x) = F_1(x+1) - 1 \ \forall x. \tag{2}$$

Finally, for completeness we also define F_0 as the identity permutation $F(x)=x$, $\forall x \in P$.

The discrimination function f and the flipping operation F define three types of pixel groups: R, S, and U depending on how the flipping changes the value of the discrimination function: G is regular if $f(F(G)) > f(G)$, G is singular if $f(F(G)) < f(G)$, and G is unchanged if $f(F(G)) = f(G)$. Here, $F(G)$ means that we apply the flipping function F to the components of the vector $G=(x_1, \ldots, x_n)$.

In general, we may wish to apply different flipping to different pixels in the group G. We can capture the assignment of flipping to pixels with a mask M, which is an n-tuple with values −1, 0, and 1. We define the flipped group $F(G)$ as $(F_{M(1)}(x_1), F_{M(2)}(x_2), \ldots, F_{M(n)}(x_n))$.

In typical images, flipping the group G will more frequently lead to an increase in the discrimination function f rather than a decrease. Thus, the total number of regular groups will be larger than the total number of singular groups. Let us denote the relative number of regular groups for a non-negative mask M as R_M (in percents of all groups) and let S_M be the relative number of singular groups. We have $R_M + S_M \le 1$ and $R_{-M} + S_{-M} \le 1$. The zero-message hypothesis of our steganalytic method is that for typical cover images, the value of R_M is approximately equal to that of R_{-M}, and the same should be true for S_M and S_{-M} :

$$R_M \cong R_{-M} \text{ and } S_M \cong S_{-M}. \tag{3}$$

We can justify this hypothesis heuristically by inspecting Equation 2. Using the flipping operation F_{-1} is the same as applying F_1 to an image whose colors have been shifted by one. Because the discrimination function f captures the smoothness, adding the value of 1 to all pixels should not influence the statistics of regular and singular groups in any significant way. Indeed, we have extensive experimental evidence that Equation 3 holds very accurately for images taken with a digital camera for both JPEG and uncompressed formats. It also holds well for images processed with common image processing operations and for scanned photographs. The relationship in Equation 3, however, is violated after randomizing the LSB plane (because of LSB steganography, for example).

Randomization of the LSB plane forces the difference between R_M and S_M to zero as the length m of the embedded message increases. After flipping the LSB of 50% of pixels (which is what would happen after embedding a random message bit into every pixel), we obtain $R_M \cong S_M$. What is surprising is that randomizing the LSB plane has the *opposite* effect on R_{-M} and S_{-M}. Their difference *increases* with the length m of the embedded message. The graph that shows R_M, S_M, R_{-M}, and S_{-M} as functions of the number of pixels with flipped LSBs appears in Figure 1 (the RS diagram). To be precise, the diagram actually shows the expected values of R_M and S_M over the statistical sample of all possible LSB randomizations, but, to simplify the notation, we use the same symbols for the expected values.

Clique type	F_1 flipping	F_{-1} flipping
$r = s = t$	$2R$, $2S$, $4U$	$8R$
$r = s > t$	$2R$, $2S$, $4U$	$4R$, $4U$
$r < s > t$	$4R$, $4S$	$4R$, $4S$
$r > s > t$	$8U$	$8U$

Table 1 Four types of cliques.

We provide a simple explanation for the peculiar increase in the difference between R_{-M} and S_{-M} for the mask M=[0 1 0].
Similar arguments could be used for other masks. We define sets $C_i = \{2i, 2i+1\}$, i=0, …, 127, and cliques of groups $C_{rst} = \{G \mid G \in C_r \times C_s \times C_t\}$. There are 128^3 cliques, each clique consisting of 8 groups (triples). The cliques are closed under LSB randomization. For the purpose of our analysis, we recognize four different types of cliques ignoring horizontally and vertically symmetrical cliques. Table 1 shows the four clique types and the number of R, S, and U groups under F_1 and F_{-1} after randomization. From the table, one can see that while randomization of LSBs has a tendency to equalize the number of R and S groups in each clique under F_1, it will increase the number of R groups and decrease the number of S groups under F_{-1}.

The principle of our new steganalytic method, which we call the RS Steganalysis, is to estimate the four curves of the RS diagram and calculate their intersection using extrapolation. We have collected experimental evidence that the R_{-M} and

S_{-M} curves are well modeled with straight lines, while second-degree polynomials can approximate the “inner” curves R_M and S_M reasonably well. We can determine the parameters of the curves from the points marked in Figure 1.

If we have a stego-image with a message of an unknown length p (in percents of pixels) embedded in the LSBs of randomly scattered pixels, our initial measurements of the number of R and S groups correspond to the points $R_M(p/2)$, $S_M(p/2)$, $R_{-M}(p/2)$, and $S_{-M}(p/2)$ (see Figure 1). The factor of one half is because – assuming the message is a random bit-stream – on average only one half of the pixels will be flipped by message embedding. If we flip the LSBs of *all* pixels in the image and calculate the number of R and S groups, we will obtain the four points $R_M(1–p/2)$, $S_M(1–p/2)$, $R_{-M}(1–p/2)$, and $S_{-M}(1–p/2)$, see Figure 1.

By randomizing the LSB plane of the stego-image, we will obtain the middle points $R_M(1/2)$ and $S_M(1/2)$. Because these two points depend on the particular randomization of the LSBs, we should repeat the process many times and estimate $R_M(1/2)$ and $S_M(1/2)$ from the statistical samples. It is possible to avoid this time consuming statistical estimation and, simultaneously, make the message length estimation much more elegant by accepting two more (natural) assumptions: (1) The point of intersection of the curves R_M and R_{-M} has the same x coordinate as the point of intersection for the curves S_M and S_{-M} (this is essentially Equation 3), and (2) $R_M(1/2) = S_M(1/2)$.

We experimentally verified the first assumption for a large database of images with unprocessed raw BMPs, JPEGs, and processed images (the second assumption could actually be proved).

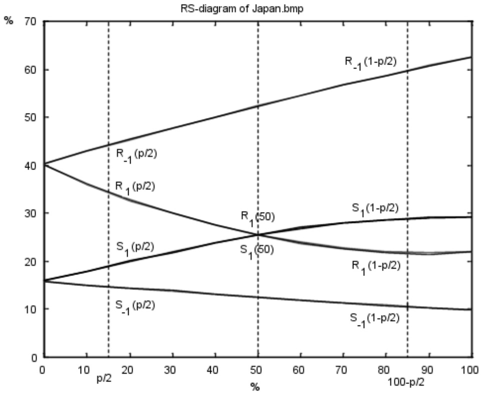

Figure 1 RS-diagram of an image taken by a digital camera. The x-axis is the percentage of pixels with flipped LSBs, the y-axis is the relative number of regular and singular groups with masks M and -M, M=[0 1 1 0].

The number of R and S groups at $p/2$ and $1–p/2$ define the straight lines, and the remaining points together with the assumptions (1) and (2) above provide enough constraints to uniquely determine the parabolas and their intersections. After rescaling the x axis so that $p/2$ becomes 0 and $100–p/2$ becomes 1, which is obtained by the linear substitution $z = (x–p/2)/(1–p)$, the x-coordinate of the intersection point can be calculated from the root of the following quadratic equation

$$2(d_1 + d_0) z^2 + (d_{-0} – d_{-1} – d_1 – 3d_0) z + d_0 – d_{-0} = 0, \text{ where}$$

$d_0 = R_M(p/2) – S_M(p/2)$, $d_1 = R_M(1–p/2) – S_M(1–p/2)$, $d_{-0} = R_{-M}(p/2) – S_{-M}(p/2)$, and $d_{-1} = R_{-M}(1–p/2) – S_{-M}(1–p/2)$.

We calculate the message length p from the root z whose absolute value is smaller by

$$p = z/(z–1/2). \tag{4}$$

There are three main factors that influence the accuracy of the estimated message length: the initial bias, the noise level or quality of the cover image, and the placement of message bits in the image.

Initial bias: The RS steganalysis may indicate a small non-zero message length due to random variations even for the original cover-image. This initial non-zero *bias* could be both positive and negative and it puts a limit on the achievable accuracy of RS steganalysis. We have tested this initial bias for a large database of 331 grayscale JPEG images and obtained a Gaussian distribution with a standard deviation of 0.5% (message length expressed in percents of the total capacity 1 bpp). Smaller images tend to have higher variation in the initial bias due to smaller number of R and S groups. Scans of half-toned images and noisy images exhibit larger variations in the bias as well. On the other hand, the bias is typically very low for JPEG images, uncompressed images obtained by a digital camera, for scans of photographs, and for images processed with typical image processing filters. As another rule of thumb, we state that color images exhibit larger variations in the initial bias than grayscales.

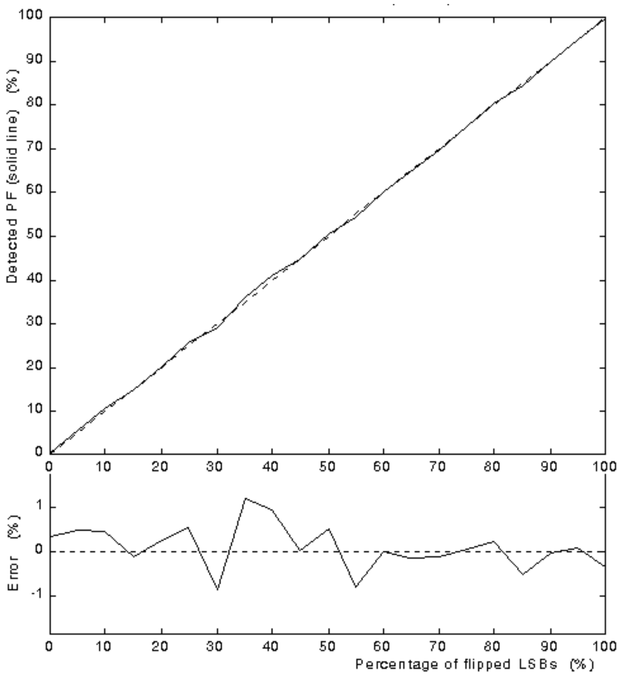

Figure 2 Estimated percentage of flipped pixels using the RS Steganalysis (solid line) vs. the actual number of flipped pixels for 'kyoto.bmp'. The bottom part of the figure shows the magnified detection error.

Noise: For very noisy images, the difference between the number of regular and singular pixels in the cover image is small. Consequently, the lines in the RS diagram intersect at a small angle and the accuracy of RS Steganalysis decreases. The same is true for low-quality images, overcompressed images, and small images (due to insufficient statistics).

Message placement: The RS Steganalysis is more accurate for messages that are randomly scattered in the stego-image than for messages concentrated in a localized area of the image. To address this issue, we could apply the same algorithm to a sliding rectangular region of the image.

Figure 2 demonstrates the extraordinary accuracy of RS steganalysis. A color 1536×1024 image 'kyoto.bmp' was taken with the Kodak DC260 digital camera, converted to grayscale, and down-sampled to 384×256 pixels. A series of stego-images was created by randomizing the LSBs of 0–100% pixels in 5% increments (pixels were randomly selected). The error between the actual and estimated percentage of flipped pixels is almost always smaller than 1% of the total image capacity.

The RS Steganalysis is applicable to most commercial steganographic software products. Examples of vulnerable programs include, for example, Steganos, Windstorm, S-Tools, and Hide4PGP. WbStego and Encrypt Pic incorporate LSB embedding into sequential pixels so it is better to use the method of Section 2.2 to analyze them. We have tested the RS steganalytic method on a small sample of images processed with these software products with different message sizes. In all cases, it readily distinguished stego-images from original cover images and the estimated message length was within a few percent of the total image capacity.

Image	Red (%)	Green (%)	Blue (%)
Initial bias	0.00 (0.00)	0.17 (0.00)	0.33 (0.00)
Steganos	2.41 (2.44)	2.70 (2.46)	2.78 (2.49)
S-Tools	2.45 (2.45)	2.62 (2.43)	2.75 (2.44)
Hide4PGP	2.44 (2.46)	2.62 (2.46)	2.85 (2.45)

Table 2 Initial bias and estimated number of pixels with flipped LSBs for the test image 'cat.bmp'. The actual numbers that should be detected in an ideal case (zero bias assumption) are indicated in parenthesis.

To test the performance of the RS Steganalysis on images obtained using current steganographic software, we used a relatively small image with a short message. A 24-bit color photograph originally stored in the JPEG format, taken by the Kodak DC260 digital camera (original resolution 1536×1024) was cropped to 1024×744 pixels. A message of length 5% of the total image capacity (100% = 3 bits per pixel) was embedded into LSBs of randomly selected pixels. The results are shown in Table 2.

The principles of RS steganalysis can be extended to variants of LSB embedding in indices of palette images and quantized DCT coefficients in JPEG files. Detailed exposition of this is subject of current and future research.

'kyoto.bmp'

'cat.bmp'

2.4. Steganalysis based on JPEG compatibility

All steganographic methods strive to achieve the minimal amount of distortion in order to minimize the likelihood of introducing detectable artifacts. However, if the cover-image, was initially stored in the JPEG format (as it is frequently the case), message embedding in the spatial domain will disturb but *not* erase the characteristic structure created by the JPEG compression and one can still easily determine whether or not a given image has been stored as JPEG in the past. Indeed, it is possible to recover the JPEG quantization table from the stego-image by carefully analyzing the values of DCT coefficients in all 8×8 blocks. After message embedding, however, the cover-image will become (with a high probability) incompatible with the JPEG format in the sense that it may be possible to prove that a particular 8×8 block of pixels could not have been produced by JPEG decompression of any block of quantized coefficients. This finding provides strong evidence that the block has been slightly modified. Indeed, it is highly suspicious to find an image stored in a lossless format that bears a strong fingerprint of JPEG compression, yet is not fully compatible with any JPEG compressed image. This can be interpreted as evidence of steganography.

By checking the JPEG compatibility of every block, we can potentially detect messages as short as one bit. And the steganalytic method will work for virtually any spatial steganographic or watermarking method, not just the LSB embedding. One can even attempt to estimate the message length and its position in the image by determining which 8×8 blocks are incompatible with JPEG compression. It is even possible to analyze the image and estimate the likely candidate for the cover-image or its blocks (the "closest" JPEG compatible image/block). This way, we may be able to identify individual pixels that have been modified.

Let us denote the *i*-th DCT coefficient of the *k*-th block as $d_k(i)$, $1 \le i \le 64$, $k = 1, \ldots, T$, where T is the total number of blocks in the image. In each block, all 64 coefficients are further quantized to integers $D_k(i)$ using the JPEG quantization matrix Q, $D_k(i) = [d_k(i)/Q(i)]$, where $[x] = integer\_round(x)$ for $0 \le x \le 255$, $[x]=0$ for $x < 0$, and $[x]=255$ for $x > 255$. The DCT coefficients $D_k(i)$ decompress to the block B

$$B = \left[B_{raw}\right] \text{ , } B_{raw} = DCT^{-1}(QD) \text{, where } QD = Q \otimes D \text{ for } \forall i. \tag{5}$$

In Eq. 5, ⊗ denotes matrix element-wise multiplication. We also dropped the block index k to simplify the notation. We note that because the JPEG compression is lossy, in general B_{orig} may not be equal to B.

If the block B has no pixels saturated at 0 or 255, we can write in the L^2 norm

$$||B_{raw}–B||^2 \le 16, \tag{6}$$

because $|B_{raw}(i)–B(i)| \le 1/2$ for all $i = 1, \ldots, 64$ due to rounding to integer values.

Suppose that we know the quantization matrix Q. Our steganalytic technique is based on the following question: Given an arbitrary 8×8 block of pixel values B, could this block have arisen through the process of JPEG decompression with the quantization matrix Q?

Denoting $QD'=DCT(B)$, we can write using the Parserval's equality $\left\|B - B_{raw}\right\|^2 = \left\|DCT(B) - DCT(B_{raw})\right\|^2 =$

$\|QD'-QD\|^2 \leq 16$. On the other hand, we can find a lower estimate for the expression $\|QD'-QD\|^2$ by substituting for $QD(i)$ the closest integer multiple of $Q(i)$:

$$16 \geq \|QD'-QD\|^2 \geq \sum_{i=1}^{64} \left| QD'(i) - Q(i) round\left(\frac{QD'(i)}{Q(i)} \right) \right| = S\,. \tag{7}$$

The quantity S can be calculated from the block B provided the quantization matrix Q is known. If S is larger than 16, we can conclude that the image block B is not compatible with JPEG compression with the quantization matrix Q. We reiterate that this is true only for blocks that do not have pixels that are saturated at 0 or 255. Indeed, the estimate (6) may not hold for blocks that have saturated pixels because the truncation at 0 and 255 can be much larger than 1/2.

If, for a given block B with unsaturated pixels, S is smaller than 16, the block B may or may not be JPEG compatible. Let $q_p(i)$, p=1, … , be integer multiples of $Q(i)$ that are closest to $QD(i)$ ordered by their distance from $QD(i)$ (the closest multiple is $q_1(i)$). In order to decide whether or not a given block B is compatible with JPEG compression with quantization table Q, we need to inspect all 64-tuples of indices $\{p(1), \ldots, p(64)\}$ for which

$$S = \sum_{i=1}^{64} \left| QD'(i) - q_{p(i)}(i) \right| \leq 16 \tag{8}$$

and verify if

$$B=[DCT^{-1}(QD)],\text{ where } QD(i)= q_{p(i)}(i). \tag{9}$$

If, for at least one set of indices $\{p(1), \ldots, p(64)\}$, the expression (5) is satisfied, the block B is JPEG compatible, otherwise it is not.

The number of 64-tuples $\{p(1), \ldots, p(64)\}$ satisfying expression (8) is always finite but it rapidly increases with increasing JPEG quality factor. For quality factors higher than 95, a large number of quantization factors $Q(i)$ are 1 or 2, and the total number of combinations of all 64 indices becomes too large to handle. We performed our experiments in Matlab on a 333 MHz Pentium II computer with 128MB memory. Once the quality factor exceeded 95, the running time became too long because Matlab ran out of memory and had to access the hard disk. We acknowledge this complexity increase as a limitation of this detection approach.

Description of the detection algorithm:

1. Divide the image into a grid of 8×8 blocks, skipping the last few rows or columns if the image dimensions are not multiples of 8.
2. Arrange the blocks in a list and remove all saturated blocks from the list (a block is saturated if it has at least one pixel with a gray value 0 or 255). Denote the total number of blocks as T.
3. Extract the quantization matrix Q from all T blocks as described below. If all the elements of Q are ones, the image was not previously stored as JPEG and our steganalytic method does not apply (exit this algorithm). If more than one plausible candidate exists for Q, the steps 4–6 need to be carried out for all candidates and the results that give the highest number of JPEG compatible blocks will be accepted as the result of this algorithm.
4. For each block B calculate the quantity S (see Equation 7).

   If S>16, the block B is not compatible with JPEG compression with quantization matrix Q.

   If S≤16, for each DCT coefficient QD_i' calculate the closest multiples of $Q(i)$, order them by their distance from QD_i', and denote them $q_p(i)$, p=1, …. For those combinations, for which the inequality (8) is satisfied, check if expression (9) holds. If, for at least one set of indices $\{p(1), \ldots, p(64)\}$ the expression (9) is satisfied, the block B is JPEG compatible, otherwise it is not.

After going through all T blocks, if no incompatible JPEG blocks are found, the conclusion is that our steganalytic method did not find any evidence for presence of secret messages. If, on the other hand, there are some JPEG incompatible blocks, we can attempt to estimate the size of the secret message, locate the message-bearing pixels, and even attempt to obtain the original cover image before secret message embedding started.

If all blocks are identified as JPEG incompatible or if the image does not appear to be previously stored as JPEG, we should repeat the algorithm for different 8×8 divisions of the image (shifted by 0 to 7 pixels in the x and y directions). This step may be necessary if the cover image has been cropped prior to message embedding.

As a consequence of this research, we strongly urge the users of steganographic programs to avoid using images that have been previously stored in the JPEG format as cover-images for steganography. The JPEG compression imprints a unique fingerprint on the image and this fingerprint can be carefully utilized for steganalysis. If no other images are available, try to rescale or resample the image to slightly smaller dimensions to wipe out the JPEG fingerprint. Applying filters, adjusting contrast, or slightly rotating the image may also help in reducing or eliminating the JPEG fingerprint.

2.4.1. Deriving the quantization matrix

Here, we show how one can extract the quantization matrix Q from a decompressed (and possibly modified) JPEG image. Keeping the same notation as in the previous section, we first calculate the DCT coefficients $d_k(i)$, $0 \le i \le 64$, $k = 1, \ldots, T$ from all unsaturated 8×8 blocks. For each coefficient i, we plot the quantity $E_i(q)$ as a function of $q = 1, 2, \ldots$

$$E_i(q) = \frac{1}{T}\sum_{k=1}^{T}\left| d_k(i) - q \times integer\_round\left(\frac{d_k(i)}{q}\right)\right| .$$

The quantity $E_i(q)$ measures the compatibility of all i-th DCT coefficients in the image with the quantization step q. If the image under inspection was indeed previously stored as JPEG, we will observe a drop (local minimum) at the correct quantization value q and at all integer divisors of q. It is intuitively clear that the correct value of the quantization step should be the largest value q at which a local minimum of E occurs. As discussed below, this may not, however, be true in all cases.

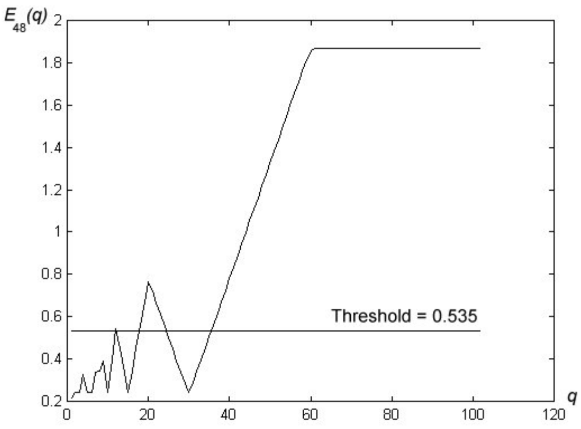

Figure 3 The error $E_{48}(q)$ for the 48-th DCT coefficient (8,6) plotted as a function of the quantization step q. Notice the local minima at 30 and all its divisors 15, 10, 6, 5, 3, and 2. The last minimum is 30, which corresponds to the correct quantization.

First of all, it is possible that some "ghost" local minima can occur for values of q larger than the correct value. Those minima, however, are significantly larger than the other “correct” minima. By setting a threshold on the largest possible value of a local minimum, we can successfully “filter out” the false minima. In our experiments, we make the threshold dependent on the DCT coefficient (on index i) and calculate its value as $\mu_i+3\sigma_i$, where μ_i and σ_i are the mean and the standard deviation of all local minima of the vector $E_i(q)$. If the vector $E_i(q)$ does not have any local minima except for q=1 (we note that $E_i(1) \le E_i(q)$ for all q>1), we inspect the difference between $E_i(1)$ and $E_i(2)$ to decide if the quantization step is 1 or 2. For this case, we developed an empirical rule that gives very good results. Based on our experiments, if $E_i(1) < 0.6\times E_i(2)$, we conclude that the quantization step is 1, otherwise it is 2.

This process of finding the quantization steps for each coefficient is still not completely foolproof. For small images, it can happen that, for the quantization step q, all values of the DCT coefficients $d_k(i)$ will be multiples of $2q$. In this case, we have no means to tell if the correct step is $2q$ or q. The probability of this happening decreases with the image size. To address this problem, we inspect the extracted quantization matrix Q for outliers by dividing it by the standard JPEG quantization table. Here, we use the logic that even customized tables (e.g., tables used in JPEG compression in digital cameras) will not significantly deviate from the standard quantization table. Finally, it is always possible to identify candidates for suspicious quantization steps and run the steganalysis for all possible combinations.

2.5. Universal blind steganalysis

With the exception of the JPEG compatibility steganalysis, which can be applied to any spatial steganographic method, all previously proposed methods are tailored to a specific embedding algorithm or its slight variations. Universal blind steganalysis is a meta-detection method in the sense that it can be adjusted, after training on original and stego-images, to detect any steganographic method regardless of the embedding domain. The trick is to find an appropriate set of sensitive statistical quantities (a feature vector) with "distinguishing" capabilities. Neural networks, clustering algorithms, and other tools of soft computing can then be used to find the right thresholds and construct the detection model from the collected experimental data.

Farid[6] proposes a set of sensitive higher-order statistics derived from the wavelet decomposition of the stego-image. Then, he uses Fisher Linear Discrimination analysis to divide the feature vectors into two linear subspaces, one corresponding to stego-images, the other to original images. The decision threshold can be adjusted to trade missed detections for false positives. Considering the fact that this approach contains a number of rather arbitrary "ad hoc" choices, it is remarkable how well his method performs.

Farid's approach starts with the n-th level wavelet decomposition of the stego-image with $V_i(x,y)$, $H_i(x,y)$, and $D_i(x,y)$ denoting the vertical, horizontal, and diagonal subbands at scale i. Then, he calculates the first four moments for all three subbands for all levels $i = 1, \ldots, n - 1$. This gives the total of $12(n - 1)$ statistical quantities. Then, he uses optimal linear predictor and collects the same statistical moments for the prediction error between the actual value of the wavelet coefficient and the optimal linear prediction from its spatial, orientation, and scale neighbors (total of 7 neighbors). For example, the vertical wavelet coefficients are predicted using the following linear predictor: $V_i(x,y) = w_1V_i(x - 1, y) + w_2V_i(x + 1, y) + w_3V_i(x, y - 1) + w_4V_i(x, y+1) + w_5V_{i+1}(x/2, y/2) + w_6D_i(x, y) + w_7D_{i+1}(x/2,y/2)$.

The weights that minimize the squared prediction error are calculated using the standard least square approach. Denoting the linear prediction for $V_i(x,y)$ as $\tilde{V}_i\ (x,y)$, the log error of the prediction is defined as $E_v(i) = \log_2(V_i) - \log_2(\tilde{V}_i\ )$. Farid calculates the first four moments of $E_v(i)$ as another part of his feature vector. The whole process is repeated for all $n - 1$ scales with horizontal and diagonal subbands. Thus, the final length of the feature vector is $12(n–1) + 4×3(n–1) = 24(n–1)$. The feature vector is calculated for a large database of original images and stego-images embedded with a fixed message size. Farid uses Fisher Linear Discriminant (FLD) analysis to classify the feature vectors into two clusters separated by a threshold. The advantage of this approach is that FLD analysis enables fast and simple scalar thresholding in one dimension.

The results reported by Farid show that his approach is a viable methodology that gives very impressive results. Farid used a database of 1400×1000 high quality images in which he embedded a small grayscale image 256×256 pixels using J-Steg, EZ Stego, and both versions of OutGuess. The best results were obtained for J-Steg with 97.8% detection reliability at 1.8% of false positives. EZ Stego was detected with 86.6% reliability with 13.2% of false positives. The same results for OutGuess were 80.4% and 19.4% (for version 0.1), and 77.7% and 23.8% for version 0.2.

We close this section with a note that methods specifically targeted to a specific steganographic method, such as the RS steganalysis, will most likely give more accurate and reliable results than any universal blind steganalytic method. Nevertheless, universal blind approaches are very important because of their flexibility and ability to be quickly adjusted to new or completely unknown steganalytic methods.

2.6. Unique fingerprints

Some steganographic techniques or their specific implementations create very unique and easily detectable artifacts. For example, the demo version of the Los Alamos scheme[19] can be quickly broken by inspecting the last row of the stego-image because this row serves as a side channel and contains information about color pairs used for embedding. Gifshuffle[22], produces images with randomized palettes, which is also a suspicious and an easy-to-check artifact. S-Tools preprocess the image palette and create clusters of very close colors that are swapped for embedding. A simple analysis of the image palette can point to the existence of secret messages. More examples can be found in the paper by Anderson[2].

3. LESSONS LEARNED

The first attempts to define the concept of security in steganography can be found in the work of Cachin[4] and Petitcolas[2]. Researchers agree that the attacker (warden) should have detailed knowledge of the stego system, including all details there are to know about the source of cover-images, but lacking the knowledge of a secret key. This so-called Kerckhoff's principle is always assumed in cryptography. However, while it is quite reasonable to assume that in cryptography all details of the encryption mechanism except for the secret key are known to the adversary, it is less plausible to assume that the adversary knows all details of the source for cover images. This is because the model of the source may not exist. For example, cover-images could be drawn randomly from the web and it is just not feasible to have a *detailed* statistical model of all images that are available on the Internet. For the same reason, the usefulness of current definitions of steganographic security is limited. Although Cachin's work gives us an exact mathematical definition, it is not clear if any practical system can be proven secure under his assumptions.

The process of embedding a secret message introduces changes to the cover image. We believe that as long as the changes can be interpreted as a result of a process that naturally happens to images, it will be very hard to distinguish cover images from stego-images. For example, it is a well-known fact that noise in CCD arrays is Gaussian[12]. Actually, this noise consists of many components, such as the thermal noise, shot noise, etc. All of those components are well modeled as an i.i.d. Gaussian noise. Since different processes and different image acquisition devices (digital cameras, scanners) have different SNRs and different noise levels, it would be presumably hard to distinguish the original image from the same image after adding a small amplitude Gaussian noise. The additional noise component could simply be a consequence of another noise source during the acquisition process. This concept of security is by no means exact but its advantage is that it enables construction of "reasonably secure" systems.

Most current methods, such as LSB embedding in the spatial domain or modifying the values of quantized DCT coefficients in the frequency (JPEG) domain (this includes the F5 algorithm[21]) tacitly assume that small, randomly placed modifications of the sample values by one are not detectable because "they are masked by the noise" commonly present in images. Unfortunately, this rather heuristic belief is quite often false. In fact, the newly emerged sensitive detection techniques described in this paper seem to suggest that any bit-replacement steganographic technique, such as LSB embedding in the spatial or frequency domain, or its derivatives, can likely never lead to secure high-capacity methods because it is hard to find a general rule that would identify pseudo-random components in images that could be substituted for messages.

A steganographic process that introduces embedding changes that mimic adding a small-amplitude Gaussian noise will necessarily be more secure than any bit-replacement technique. Because during the image acquisition process, many different independent sources of Gaussian noise with varying amplitudes are superimposed onto the image, we claim that any technique that embeds message bits in the image by adding small amplitude Gaussian noise must be substantially more secure than any bit-replacement technique. This is because it is hard to establish whether the additional Gaussian noise is due to the sensor properties or steganography. Examples of high-capacity steganographic methods whose embedding artifacts mimic Gaussian noise adding are due to Marvel[16] and Alturki[1].

Obviously, the battle between steganography and steganalysis is never-ending. New, more sophisticated steganographic methods will require more refined approach for detection. Targeted detection will always provide more reliable and accurate results than universal blind detection. Nevertheless, universal blind detectors form an extremely important research direction because of their flexibility to adjust to new or unknown steganographic mechanisms (when Kerckhoff's principle does not apply). In the future, we will likely see more work addressing the important issue of "safe" capacity estimates for steganographic methods. At the same time, more accurate and sensitive detection schemes may find its way into virus-checking programs and to software products used by forensic experts and law enforcement. The existing steganalytic methods form a very small tip of a very big iceberg that hides exciting and challenging research problems for many years to come.

ACKNOWLEDGEMENTS

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under a research grant number F30602-00-1-0521. 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.

REFERENCES

1. F. Alturki and R. Mersereau, “A Novel Approach for Increasing Security and Data Embedding Capacity in Images for Data Hiding Applications”, *Proc. ITCC*, Las Vegas, NV, April 2–4, 2001, pp. 228–233.
2. R.J. Anderson and F.A.P. Petitcolas, "On the limits of steganography", *IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection*, **16(**4), pp. 474–481, 1998.
3. T. Aura, “Practical invisibility in digital communication,” Lecture Notes in Computer Science, vol.1174, Springer-Verlag, 1996, pp. 265–278.
4. C. Cachin, “An Information-Theoretic Model for Steganography”, *Lecture Notes on Computer Science*, vol. 1525, Springer-Verlag, New York, 1998, pp. 306–318.
5. R. Chandramouli and N. Memon, "Analysis of LSB based Image Steganography Techniques", *Proceedings of ICIP 2001*, Thessaloniki, Greece, October 7–10, 2001.
6. H. Farid, “Detecting Steganographic Message in Digital Images”, Report TR2001-412, Dartmouth College, Hanover, NH, 2001.
7. J. Fridrich, M. Goljan, and R. Du, “Distortion-free Data Embedding”, In: *Lecture Notes in Computer Science*, vol.2137, Springer-Verlag, Berlin, 2001.
8. J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compatibility," *SPIE Multimedia Systems and Applications IV*, Denver, CO, August 20–24, 2001.
9. J. Fridrich, R. Du, and L. Meng, “Steganalysis of LSB Encoding in Color Images,” *Proceedings IEEE International Conference on Multimedia and Expo*, July 30–August 2, 2000, New York City, NY.
10. J. Fridrich, M. Goljan, and R. Du, "Reliable Detection of LSB Steganography in Grayscale and Color Images ", *Proc. ACM,* Special Session on Multimedia Security and Watermarking, Ottawa, Canada, October 5, 2001, pp. 27–30.
11. J. Fridrich, M. Goljan, and R. Du, “Detecting LSB Steganography in Color and Gray-Scale Images”, *Magazine of IEEE Multimedia*, *Special Issue on Security*, October-November issue, 2001, pp. 22–28.
12. G. E. Healey and R. Kondepudy, “Radiometric CCD Camera Calibration and Noise Estimation”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. **16**(3), March 1994, pp. 267–276.
13. N. F. Johnson and S. Jajodia, “Steganography: Seeing the Unseen,” *IEEE Computer*, February 1998, pp.26–34.
14. N. F. Johnson and S. Jajodia, “Steganalysis of Images Created Using Current Steganography Software,” Lecture Notes in Computer Science, vol.1525, Springer-Verlag, Berlin, 1998, pp. 273–289.
15. S. Katzenbeisser and F.A.P. Petitcolas, “On Defining Security in Steganographic Systems”, Proc. Electronic Imaging, Photonics West, SPIE 2002, San Jose, California, January (2002), submitted.
16. L.M. Marvel, C.G. Boncelet, and C.T. Retter, “Reliable Blind Information Hiding for Images”, *Lecture Notes on Computer Science*, vol. 1525, Springer-Verlag, New York, 1998, pp. 48–61.
17. N. Provos, “Defending Against Statistical Steganalysis”, *10th USENIX Security Symposium*, Washington, DC, 2001.
18. N. Provos and Peter Honeyman, “Detecting Steganographic Content on the Internet”, *CITI Technical Report 01-11,* August 2001, submitted for publication.
19. M. T. Sandford II, J. N. Bradley, and T. G. Handel, “The data embedding method”, In *Proc. of the SPIE Photonics East Conference*, Philadelphia, September 1995.
20. A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," Lecture Notes in Computer Science, vol.1768, Springer-Verlag, Berlin, 2000, pp. 61–75.
21. A. Westfeld, “High Capacity Despite Better Steganalysis (F5–A Steganographic Algorithm)”, to appear in Lecture Notes in Computer Science, vol.2137, Springer-Verlag, Berlin, 2001.
22. Steganography software for Windows, http://members.tripod.com/steganography/stego/software.html