

Reliable Detection of LSB Steganography in Color and Grayscale Images

Jessica Fridrich
SUNY Binghamton
fridrich@binghamton.edu

Miroslav Goljan
SUNY Binghamton
mgoljan@binghamton.edu

Rui Du
SUNY Binghamton
rdu@binghamton.edu

1. ABSTRACT

We describe a reliable and accurate method for detecting least significant bit (LSB) non-sequential embedding in digital images. The secret message length is derived by inspecting the lossless capacity in the LSB and shifted LSB plane. An upper bound of 0.005 bits/pixel was experimentally determined for safe LSB embedding.

1.1 Keywords

Steganalysis, steganography, LSB embedding, attacks

2. Motivation

Steganography is the art of secret communication. Its purpose is to hide the very presence of communication as opposed to cryptography which aims to make communication unintelligible to those who do not possess the right keys [2]. We can use digital images, videos, sound files, and other computer files that contain perceptually irrelevant or redundant information as “covers” or carriers to hide secret messages. After embedding a secret message into the cover-image, we obtain a so-called stego-image. It is important that the stego-image does not contain any detectable artifacts due to message embedding. A third party could use such artifacts as an indication that a secret message is present. Once a third party can reliably identify which images contain secret messages, the steganographic tool becomes useless.

Obviously, the less information we embed into the cover-image, the smaller the probability of introducing detectable artifacts by the embedding process. Another important factor is the choice of the cover-image. The selection is at the discretion of the person who sends the message. Images with a low number of colors, computer art, images with a unique semantic content, such as fonts, should be avoided as cover images. Some steganographic experts recommend grayscale images as the best cover-images [3]. They recommend uncompressed scans of pho-

tographs or images obtained with a digital camera containing a high number of colors and consider them safe for steganography.

In previous work [4], we have shown that images stored previously in the JPEG format are a very poor choice for cover images. This is because the quantization introduced by JPEG compression can serve as a “watermark” or a unique fingerprint, and you can detect even very small modifications of the cover image by inspecting the compatibility of the stego-image with the JPEG format.

In [5], we developed a steganographic method for detection of LSB embedding in 24-bit color images (the Raw Quick Pairs – RQP method). We based it 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.

Pfitzmann and Westfeld [6] introduced a method based on statistical analysis of Pairs of Values (PoVs) that are exchanged during message embedding. 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). However, we can only detect randomly scattered messages with this method when the message length becomes comparable with the number of pixels in the image.

Johnson and Jajodia [7,8] pointed out that steganographic methods for palette images that preprocess the palette before embedding are very vulnerable. Several steganographic programs 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 (clusters) of colors that can be easily detected.

3. Lossless data embedding

In our previous work on lossless (or invertible) data embedding [1], we proposed an idea for a new steganalytic method for detection of LSB embedding in color and gray-scale images. The method originated by analyzing the capacity for lossless data embedding in the LSBs. Randomizing the LSBs decreases the lossless capacity in the LSB plane, but it has a different influence on the capacity for embedding that is not constrained to one bit-plane. Thus, the lossless capacity turned out to be a very sensitive measure for the degree of randomization of the LSB plane. Note that for most images the LSB plane is essentially random and does not contain any easily recognizable structure. Using classical statistical quantities constrained to the LSB plane to capture the degree of randomization is very unreliable. The lossless capacity reflects the fact that the LSB plane, even though it looks random, is never the less related to the other bit-planes. This relationship, however, is not linear but nonlinear, and the lossless capacity seems to measure this relationship fairly well. This is why we proposed it for steganography detection.

To explain the details of our new steganalytic technique, we will first briefly explore the main paradigms behind lossless embedding.

Let us assume that we have a cover image with $M \times N$ pixels and with pixel values from the set P . For example, for an 8-bit grayscale image, $P = \{0, \dots, 255\}$. The lossless embedding starts with dividing the image into disjoint groups of n adjacent pixels (x_1, \dots, x_n) . As an example, we can choose groups of $n=4$ consecutive pixels in a row. We further define so called discrimination function f that assigns a real number $f(x_1, \dots, x_n) \in \mathbf{R}$ to each pixel group $G = (x_1, \dots, x_n)$. The purpose of the discrimination function is to capture the smoothness or "regularity" of the group of pixels G . The noisier the group of pixels $G=(x_1, \dots, x_n)$ is, the larger the value of the discrimination function becomes. For example, we can choose the 'variation' of the group of pixels (x_1, \dots, x_n) as the discrimination function f :

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

We can use image models or statistical assumptions about the cover image for the design of other discrimination functions.

Finally, we define an invertible operation F on P called *flipping*. Flipping will be a permutation of gray levels that entirely consists of 2-cycles. Thus, F will have the property that $F^2 = \text{Identity}$ or $F(F(x)) = x$ for all $x \in P$. The permutation $F_1: 0 \leftrightarrow 1, 2 \leftrightarrow 3, \dots, 254 \leftrightarrow 255$ corresponds to flipping (negating) the LSB of each gray level. We further define shifted LSB flipping F_{-1} as $-1 \leftrightarrow 0, 1 \leftrightarrow 2, 3 \leftrightarrow 4, \dots, 253 \leftrightarrow 254, 255 \leftrightarrow 256$, or

$$F_{-1}(x) = F_1(x+1) - 1 \text{ for all } x. \quad (2)$$

For completeness, we also define F_0 as the identity permutation $F(x)=x$ for all $x \in P$. We use the discrimination function f and the flipping operation F to define three types of pixel groups: R, S , and U

$$\underline{\text{Regular groups:}} \quad G \in R \Leftrightarrow f(F(G)) > f(G)$$

$$\underline{\text{Singular groups:}} \quad G \in S \Leftrightarrow f(F(G)) < f(G)$$

$$\underline{\text{Unusable groups:}} \quad G \in U \Leftrightarrow f(F(G)) = f(G).$$

In the expressions above, $F(G)$ means that we apply the flipping function F to the components of the vector $G=(x_1, \dots, x_n)$. 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 $(F_{M(1)}(x_1), F_{M(2)}(x_2), \dots, F_{M(n)}(x_n))$. The purpose of the flipping F is perturbing the pixel values in an invertible way by some small amount thus simulating the act of invertible noise adding. In typical pictures, adding small amount of noise (i.e., flipping by a small amount) will lead to an increase in the discrimination function rather than a decrease. Thus, the total number of regular groups will be larger than the total number of singular groups. This bias allows for lossless imperceptible embedding of a potentially large amount of information (for more details, see [1]).

4. Steganalytic technique

Let us denote the number of regular groups for mask M as R_M (in percents of all groups). Similarly, S_M will denote the relative number of singular groups. We have $R_M + S_M \leq 1$ and $R_{-M} + S_{-M} \leq 1$, for the negative mask. The statistical hypothesis of our steganalytic method is that in a typical image, the expected value of R_M is equal to that of R_{-M} , and the same is true for S_M and S_{-M} :

$$R_M \cong R_{-M} \text{ and } S_M \cong S_{-M}. \quad (3)$$

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. For a typical image, there is no a priori reason why the number of R and S groups should change significantly by shifting the colors by one.

Indeed, we have extensive experimental evidence that Equation 3 holds very accurately for images taken with a digital camera for both lossy and lossless formats. It also holds well for images processed with common image processing operations and for most scanned images. 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 $R_M \cong S_M$. This is like saying that the lossless embedding capacity in the LSB

plane is zero [1]. What is surprising is that the influence of 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).

We have a simple explanation for the peculiar increase in the difference between R_{-M} and S_{-M} for the mask $M=[0\ 1\ 0]$. We define sets $C_i = \{2i, 2i+1\}$, $i=0, \dots, 127$, 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. The table below shows the four types and the number of R , S , and U groups under F_1 and F_{-1} for each type. 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} .

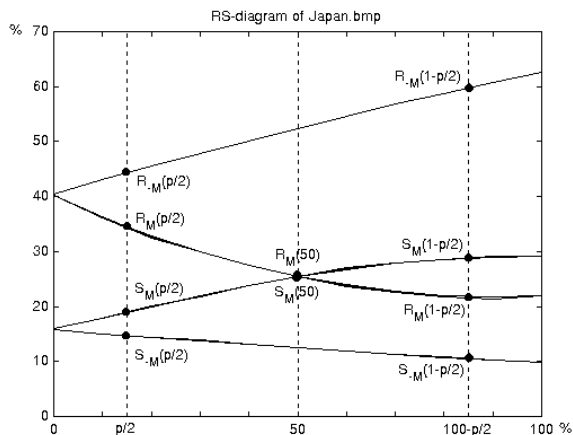


Figure 1 RS-diagram of an image taken by a digital camera. The x -axis is the percentage of pixels with flipped LSBs, the y -axis is the relative number of regular and singular groups with masks M and $-M$, $M=[0\ 1\ 0]$.

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

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. The general shape of the four curves in the diagram varies with the cover-image from almost perfectly linear to curved. 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. (Part of our future effort is a theoretical explanation of their shapes.) 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. We can fit straight lines through the points $R_{-M}(p/2)$, $R_{-M}(1-p/2)$ and $S_{-M}(p/2)$, $S_{-M}(1-p/2)$. The points $R_M(p/2)$, $R_M(1/2)$, $R_M(1-p/2)$, and $S_M(p/2)$, $S_M(1/2)$, $S_M(1-p/2)$ determine two parabolas. Each parabola and a corresponding line intersect to the left. The arithmetic average of the x coordinates of both intersections lets us estimate the unknown message length p .

We can avoid the time consuming statistical estimation of the middle points $R_M(1/2)$ and $S_M(1/2)$ 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 a stronger version of Equation 3.
2. The curves R_M and S_M intersect at $m=50\%$, or $R_M(1/2) = S_M(1/2)$. This assumption is like saying that the lossless embedding capacity for a randomized LSB plane is zero.

We experimentally verified these assumptions for a large database of images with unprocessed raw BMPs, JPEGs, and processed BMP images. The two assumptions make it possible to derive a simple formula for the secret message length p . After rescaling the x axis so that $p/2$ becomes 0 and $100-p/2$ becomes 1, the x -coordinate of the intersection point is a root of the following quadratic equation

$$2(d_1 + d_0)x^2 + (d_{-0} - d_{-1} - d_1 - 3d_0)x + d_0 - d_{-0} = 0,$$

where

$$d_0 = R_M(p/2) - S_M(p/2), \quad d_1 = R_M(1-p/2) - S_M(1-p/2),$$

$$d_{-0} = R_{-M}(p/2) - S_{-M}(p/2), \quad d_{-1} = R_{-M}(1-p/2) - S_{-M}(1-p/2).$$

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

$$p = x/(x-1/2). \quad (4)$$

Because of space limitations, we omit the derivation of these equations. Suffice it to say that the number of R and S groups at $p/2$ and $1-p/2$ define the straight lines, and the assumptions 1 and 2 provide enough constraints to uniquely determine the parabolas and their intersections.

4.1 Accuracy

We can use Equation 4 to estimate the size of the secret message embedded in the stego-image. The initial bias, the noise level of the cover image, and the placement of message bits in the image are the three main factors that influence the accuracy of the estimated message length.

Initial bias: Even original cover-images may indicate a small non-zero message length due to random variations. This initial non-zero bias could be both positive and negative and it puts a limit on the theoretical accuracy of our steganalytic method. 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% (see Figure 2). Smaller images tend to have higher variation in the initial bias because of the 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, and high resolution scans. As another rule of thumb, we state that color images exhibit larger variation in the initial bias than grayscales.

If we can estimate the initial message length ml_0 (the bias), we can use the following formula to correct the detected message length ml_{det}

$$ml = \frac{ml_{det} - ml_0}{1 - ml_0} \quad (5)$$

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 the RS Steganalysis decreases.

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 can apply the same algorithm to a sliding rectangular region of the image. For sequentially embedded messages, the method described in [6] is also a good alternative.

5. Experimental results

In our first test, we used the Kodak DC260 digital camera and converted a color 1536×1024 image ‘kyoto.bmp’ to grayscale and down-sampled to 384×256 pixels. We created a series of stego-images from the original image by randomizing the LSBs of 0–100% pixels in 5% increments. Using our method, we detected the number of pixels with

flipped LSBs in each stego-image (we used groups of 2×2 pixels with the mask [1 0; 0 1]). The result, typical for images with an initial bias close to zero, is plotted in Figure 3. As can be seen from the chart, the error between the actual and estimated percentage of flipped pixels is almost always smaller than 1%.

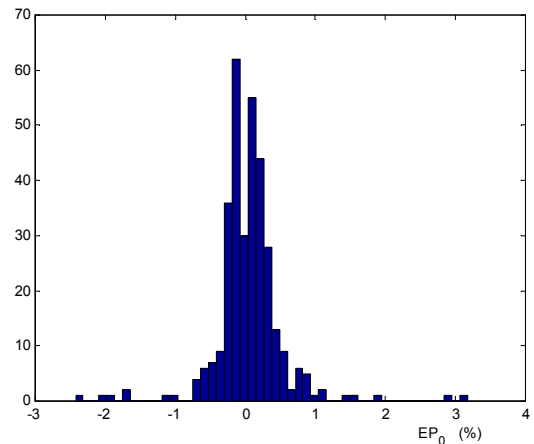


Figure 2 Histogram of the initial bias (in percents of the total number of pixels) in 331 original cover images of size 250×350 pixels stored in the JPEG format.

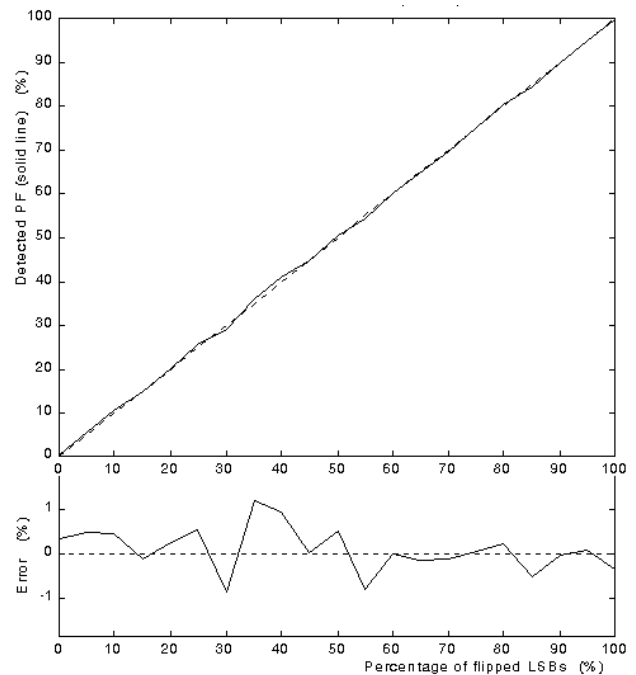


Figure 3 Estimated percentage of flipped pixels using the RS Steganalysis (solid line) vs. the actual number of flipped pixels for ‘kyoto.bmp’. The bottom part of the figure shows the magnified detection error.

The RS Steganalysis is applicable to most commercial steganographic software products (to see some of the steganography software available for Windows, you might want to check out <http://members.tripod.com/steganography/stego/software.html>). 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 described in [6] 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 off the actual message length.

StegoDos and Hide&Seek use LSB embedding in indices to palette entries (for palette images or GIFs). Although testing our RS steganography for palette images remains a part of our future work, we believe that similar concepts are equally applicable to GIFs with randomly scattered messages.

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. The test image was a scanned color photograph 422×296 and the message was a random bit sequence with 375 kb or 20% of the image full capacity (100% = 3bits per pixel). Since the initial bias is about 2.5% in each color channel (see Table 1), as indicated in the first row of Table 2, according to Equation 5 the expected detected percentage of flipped pixels would be about 12.25%.

Image	Red (%)	Green (%)	Blue (%)
Cover image	2.5 (0.0)	2.4 (0.0)	2.6 (0.0)
Steganos	10.6 (9.8)	13.3 (9.9)	12.4 (9.8)
S-Tools	13.4 (10.2)	11.4 (10.2)	10.3 (10.2)
Hide4PGP	12.9 (10.0)	13.8 (10.1)	13.0 (10.0)

Table 1 Initial bias and estimated number of pixels with flipped LSBs for the test image 'siesta.bmp'. The actual numbers that should be detected in an ideal case (zero bias assumption) are indicated in parenthesis.

As another test, we took a 24-bit color photograph originally stored in the JPEG format, taken by the Kodak DC260 digital camera (original resolution 1536×1024) cropped to 1024×744 pixels, with a very short embedded message of length 5% (100% = 3 bits per pixel). The results shown in Table 2 demonstrate the extraordinary accuracy of the RS Steganalysis.

Image	Red (%)	Green (%)	Blue (%)
Cover image	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.

Test images:



'kyoto.bmp'



'siesta.bmp'



'cat.bmp'

6. Conclusions and future directions

Steganography is a tool using which the very act of communication can be concealed. In combination with cryptography, it provides a very secure mode of communication. While privacy is an important aspect of our lives, steganography can be and has already been misused. Recently, the US Today printed an article "Terror groups hide behind Web encryption" by Jack Kelley, USA TODAY (04/13/2001 - Updated 04:41 PM ET). In his article, Mr. Kelley writes: "... U.S. officials and experts say [steganography] is the latest method of communication being used by Osama bin Laden and his associates to outfox law enforcement. All the Islamists and terrorist groups are now using the Internet to spread their messages," says Reuven Paz, academic director of the Institute for Counter-Terrorism, an independent Israeli think tank. The Internet has proven to be a boon for terrorists." A full version of the article can be found at <http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen.htm>.

The importance of techniques that can reliably detect the presence of secret messages in images is increasing. Images can hide a large amount of malicious code that could be activated by a small Trojan horse type of virus. Indeed, we believe that detection of hidden information in images should be a part of every virus-detection software. Because most of currently available software packages employ a form of LSB embedding information, we believe that the new RS Steganalysis is an important contribution that will find industrial numerous applications for law enforcement and industry in general.

The experimental results obtained by RS Steganalysis also provide a new estimate on “safe” size of secret messages embedded using LSB embedding. For high quality images from scanners and digital cameras, we estimate that messages requiring less than 0.005 bits per pixel are undetectable using RS Steganalysis. Higher bit rates are in the range of detectability using RS Steganalysis.

We’re focusing our future research on applying RS Steganalysis for palette images. We’re also studying the possibility of estimating the initial bias from stego-images to improve the sensitivity of the RS detection method to short messages in digital images.

7. Acknowledgements

The work on this paper was partially supported by Air Force Research Laboratory, Air Force Material Command, USAF, under a research grant number F30602-00-1-0521 and partially by the AFOSR grant No. F496200110123. 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, the AFOSR, or the U. S. Government.

8. References

- [1] J. Fridrich, M. Goljan and R. Du, “Distortion-free Data Embedding”, to appear in Lecture Notes in Computer Science, vol.2137, Springer-Verlag, Berlin, 2001.
- [2] R.J. Andersen and Petitcolas, F.A.P., "On the limits of steganography," *IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection* **16** No.4 (1998) 474–481.
- [3] T. Aura, “Practical invisibility in digital communication,” Lecture Notes in Computer Science, vol.1174, Springer-Verlag, 1996, pp. 265-278.
- [4] J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compatibility," *SPIE Multimedia Systems and Applications IV*, Denver, CO, August 20–24, 2001.
- [5] J. Fridrich, R. Du, and L. Meng, “Steganalysis of LSB Encoding in Color Images,” *Proceedings IEEE International Confer-*

ence on Multimedia and Expo, July 30–August 2, 2000, New York City, NY, pp. ??-??.

[6] A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," Lecture Notes in Computer Science, vol.1768, Springer-Verlag, Berlin, 2000, pp. 61-75.

[7] N. F. Johnson and S. Jajodia, “Steganography: Seeing the Unseen,” *IEEE Computer*, February 1998, pp.26–34.

[8] 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.