Detecting LSB Steganography in Color and Gray-Scale Images

Jessica Fridrich, Miroslav Goljan, and Rui Du
State University of New York, Binghamton

We describe a reliable and accurate method for detecting least significant bit (LSB) nonsequential 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 per pixel was experimentally determined for safe LSB embedding.

Steganography is the art of secret communication. Its purpose is to hide the presence of communication, as opposed to cryptography, which aims to make communication unintelligible to those who don't possess the right keys.[1] 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's important that the stego-image doesn't 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, and images with unique semantic content (such as fonts) should be avoided as cover images. Some steganographic experts recommend grayscale images as the best cover images.[2] 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.

In previous work,[3] we've shown that images stored previously in the JPEG format are a poor choice for cover images. This is because the quantization introduced by JPEG compression can serve as a watermark or unique fingerprint, and you can detect even small modifications of the cover image by inspecting the compatibility of the stego-image with the JPEG format.

In Fridrich et al.,[4] we developed a steganographic method for detecting 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 percent 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 can't be applied to grayscale images.

Pfitzmann and Westfeld[5] introduced a method based on statistical analysis of pairs of values (PoVs) exchanged during message embedding. Pairs of colors that differ in the LSB only, for example, could form these PoVs. This method provides 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[6,7] 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 others can detect easily.

Lossless data embedding

In our previous work on lossless (or invertible) data embedding,[8] we proposed an idea for a new steganalytic method for detection of LSB embedding in color and grayscale images. The method originated by analyzing the capacity for lossless data embedding in the LSBs. Randomizing the

1070-986X/01/$10.00 © 2001 IEEE

LSBs decreases the lossless capacity in the LSB plane, but it has a different influence on the capacity for embedding that isn't constrained to one bit plane. Thus, the lossless capacity became a sensitive measure for the degree of randomization of the LSB plane. Note that for most images the LSB plane is essentially random and doesn't contain any easily recognizable structure. Using classical statistical quantities constrained to the LSB plane to capture the degree of randomization is unreliable. The lossless capacity reflects the fact that the LSB plane—even though it looks random—is related nonetheless to the other bit planes. This relationship, however, is 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'll first briefly explore the main paradigms behind lossless embedding.

Let's 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, $P = \{0, \ldots, 255\}$. The lossless embedding starts with dividing the image into disjoint groups of n adjacent pixels $(x_1, \ldots, x_n)$. As an example, we can choose groups of $n = 4$ consecutive pixels in a row. We further define a so-called discrimination function f that assigns a real number $f(x_1, \ldots, x_n) \in \mathbf{R}$ to each pixel group $G = (x_1, \ldots, x_n)$. 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 $G = (x_1, \ldots, x_n)$, the larger the value of the discrimination function becomes. For example, we can choose the variation of the group of pixels $(x_1, \ldots, x_n)$ as the discrimination function f:

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

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

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

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:

Regular groups: $G \in R \Leftrightarrow f(F(G)) > f(G)$
Singular groups: $G \in S \Leftrightarrow f(F(G)) < f(G)$
Unusable groups: $G \in U \Leftrightarrow f(F(G)) = f(G)$

In these expressions, $F(G)$ means that we apply the flipping function F to the components of the vector $G = (x_1, \ldots, x_n)$. 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 $(F_{M(1)}(x_1), F_{M(2)}(x_2), \ldots, F_{M(n)}(x_n))$. 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 a small amount of noise (for example, 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 Fridrich et al.[8]).

Steganalytic technique

Let's denote the number of regular groups for mask M as R_M (percent 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 equals that of R_{-M}, and the same is true for S_M and S_{-M}

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

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. For a typical image, there's 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 the hypothesis in 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

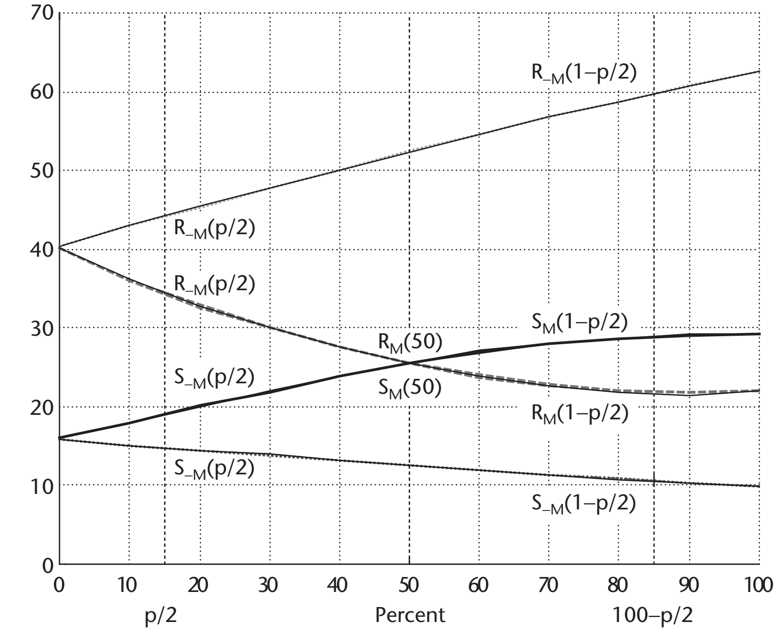


Figure 1. RS-diagram of an image taken by a digital camera. The x-axis is the percentage of pixels with flipped LSBs, the y-axis is the relative number of regular and singular groups with masks M and –M, M = [0 1 1 0].

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 percent 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.[8] What's 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 $C_i = \{2i, 2i + 1\}$, $i = 0, \ldots, 127$, and cliques of groups $C_{rst} = \{G \mid G \in C_r \times C_s \times C_t\}$. There exist 128^3 cliques, each clique consisting of eight 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 types and the number of R, S, and U groups under F_1 and F_{-1} for each type. From the table, you 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}.

Table 1. Four different types of cliques.

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 Figure 1 varies with the cover image from almost perfectly linear to curved. We've collected experimental evidence that the R_{-M} and S_{-M} curves are modeled well 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 percent 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 message embedding will flip only one half of the pixels.

If we flip the LSBs of all pixels in the image and calculate the number of R and S groups, we'll obtain the four points $R_M(1 - p/2)$, $S_M(1 - p/2)$, $R_{-M}(1 - p/2)$, and $S_{-M}(1 - p/2)$ in Figure 1. By randomizing the LSB plane of the stego-image, we 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 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 $m = 50$ percent, 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

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

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

$$p = x/(x - 1/2) \tag{4}$$

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 in Assumptions 1 and 2 provide enough constraints to uniquely determine the parabolas and their intersections.

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 nonzero message length due to random variations. This initial nonzero bias could be both positive and negative and puts a limit on the theoretical accuracy of our steganalytic method. We 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 percent (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 low for JPEG images, uncompressed images obtained by a digital camera, and high resolution scans. As another rule of thumb, color images exhibit larger variation in the initial bias than grayscales.

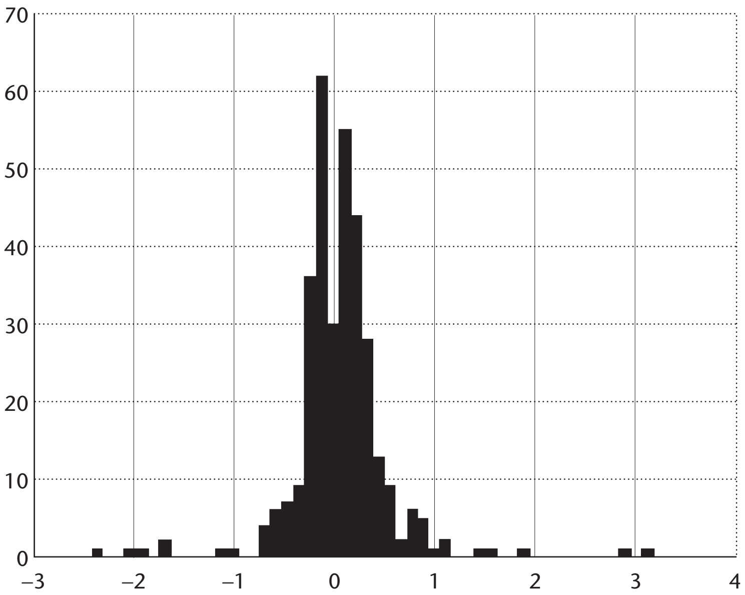


Figure 2. Histogram of the initial bias (percent of the total number of pixels) in 331 original cover images of size 250 × 350 pixels stored in the JPEG format.

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}:

$$ml = \frac{ml_{\text{det}} - ml_0}{1 - ml_0} \tag{5}$$

Noise. For 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

Figure 3. The test image kyoto.bmp used to test the RS Steganalysis' performance.

issue, we can apply the same algorithm to a sliding rectangular region of the image. For sequentially embedded messages, the method described in Fridrich et al.[5] is also a good alternative.

Experimental results

In our first test, we used the Kodak DC260 digital camera and converted a color 1536 × 1024 image to grayscale and downsampled to 384 × 256 pixels (Figure 3). We created a series of stego-images from the original image by randomizing the LSBs of 0 to 100 percent pixels in 5 percent increments. Using our method, we detected the number of pixels with flipped LSBs in each stego-image (for groups of of 2 × 2 pixels with the mask [1 0; 0 1]). We plotted the result in Figure 4, which is typical for images with an initial bias close to zero. As the chart shows, the error between the actual and estimated percentage of flipped pixels is almost always smaller than 1 percent.

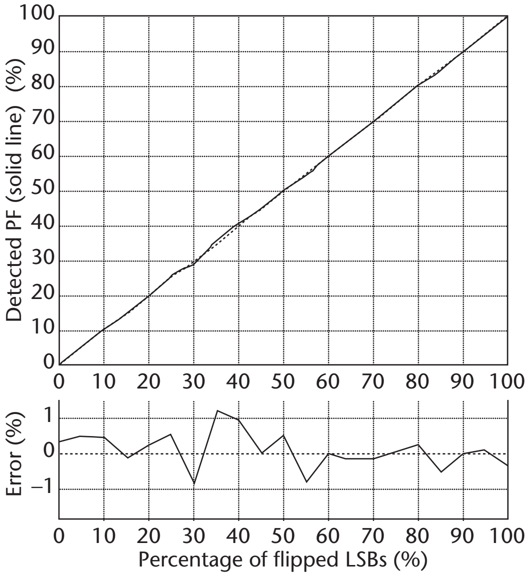


Figure 4. Estimated percentage of flipped pixels using the RS Steganalysis (solid line) versus the actual number of flipped pixels for Figure 3. The bottom part of the figure shows the magnified detection error.

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 Steganos, Windstorm, S-Tools, and Hide4PGP. WbStego and Encrypt Pic incorporate LSB embedding into sequential pixels, so it's better to use the method described in Westfeld and Pfitzmann[5] to analyze them. We 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 actual message length.

StegoDos (public domain software by Black Wolf) and Hide&Seek (freeware by Allan Latham) 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 (Figure 5) with a short message. The test image was a scanned color photograph 422 × 296 and the message was a random bit sequence with 375 Kbytes or 20 percent of the image full capacity (100 percent equals 3 bits per pixel). Since the initial bias is about 2.5 percent 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 percent.

As another test, we took a 24-bit color photograph (Figure 6) originally stored in the JPEG format, taken by the Kodak DC260 digital camera (original resolution 1536 × 1024) cropped to 1024 × 744 pixels, with a short embedded message of length 5 percent (100 percent equals 3 bits per pixel). The results in Table 3 demonstrate the accuracy of the RS Steganalysis.

Conclusions and future directions

Steganography is a tool for concealing the very act of communication. In combination with cryp-

Figure 5. The test image siesta.bmp used to test the RS Steganalysis' performance.

tography, 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 *USA Today* printed an article, "Terror Groups Hide behind Web Encryption," by Jack Kelley (printed 19 June 2001 and updated 5:05 p.m. eastern time). 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 this 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 software packages currently available employ a form of LSB embedding information, we believe that the new RS Steganalysis is an important contribution that will find numerous applications for law enforcement and industry in general.

Table 2. Initial bias and estimated number of pixels with flipped LSBs for the test image in Figure 5. The actual numbers that should be detected in an ideal case (zero bias assumption) are in parenthesis.

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)

Figure 6. The test image cat.bmp image used to test the RS Steganalysis' performance.

The experimental results obtained by RS Steganalysis also provide a new estimate on the 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. **MM**

Table 3. Initial bias and estimated number of pixels with flipped LSBs for the test image in Figure 6. The actual numbers that should be detected in an ideal case (zero bias assumption) are in parenthesis.

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)

Acknowledgments

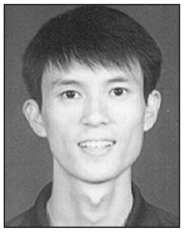
The Air Force Research Laboratory, Air Force Material Command, USAF, partially supported our research under grant number F30602-00-1-0521. An Air Force Office of Scientific Research (AFOSR) grant F496200110123 also partially supported our research. The views and conclusions contained herein are ours and shouldn't be interpreted as necessarily representing the official policies, either expressed or implied, of the Air Force Research Laboratory, the AFOSR, or the US Government.

References

1. R.J. Andersen and F.A.P. Petitcolas, "On the Limits of Steganography," *IEEE J. Selected Areas in Comm.*, vol. 16, no. 4, 1998, pp. 474-481.
2. T. Aura, "Practical Invisibility in Digital Communication," *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, vol. 1174, 1996, pp. 265-278.
3. J. Fridrich, M. Goljan, and R. Du, "Steganalysis Based on JPEG Compatibility," *SPIE Multimedia Systems and Applications IV*, SPIE Press, Bellingham, Wash., 2001.
4. J. Fridrich, R. Du, and L. Meng, "Steganalysis of LSB Encoding in Color Images," *Proc. IEEE Int'l Conf. Multimedia and Expo*, CD-ROM, IEEE Press, Piscataway, N.J., 2000.
5. A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," *Lecture Notes in Computer Science*, vol. 1768, Springer-Verlag, Berlin, 2000, pp. 61-75.
6. N.F. Johnson and S. Jajodia, "Steganography: Seeing the Unseen," *Computer*, vol. 31, no. 2, Feb. 1998, pp. 26-34.
7. 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.
8. J. Fridrich, M. Goljan, and R. Du, "Distortion-Free Data Embedding," to be published in *Lecture Notes in Computer Science,* vol. 2137, Springer-Verlag, Berlin, 2001.

Jessica Fridrich is a research professor at the Center for Intelligent Systems at State University of New York, Binghamton. In 1987, she received her MS degree in applied mathematics from Czech Technical University in Prague, Czech Republic, and her PhD in systems science in 1995 from the State University of New York in Binghamton. Her main research interests are in the field of steganography and steganalysis, digital watermarking, authentication and tamper detection, and forensic analysis of digital images.

Miroslav Goljan is a PhD candidate in the Department of Electrical engineering at SUNY Binghamton. He received his MS in theoretical informatics from Charles University in Prague in 1984. His most recent contributions include the new paradigms of lossless watermarking for images, self-embedding, and accurate LSB embedding detection.



Rui Du is a PhD candidate in the Department of Electrical Engineering at SUNY Binghamton. He received his MS in electrical engineering from Sichuan University, China, in 1998. His main research interests are in digital video compression and watermarking. As part of his PhD thesis, he created a Windows application (SecureStego) that contains many unique algorithms developed at Binghamton University, including lossless authentication of images and MPEG-2 videos, steganalysis, and lossless data embedding for all image formats.

Readers may contact Fridrich at the Center for Intelligent Systems, SUNY Binghamton, Binghamton, NY 13902-6000, email fridrich@binghamton.edu.