YASS: Yet Another Steganographic Scheme that Resists Blind Steganalysis

Kaushal Solanki[††], Anindya Sarkar[†], and B. S. Manjunath[†]

†Department of Electrical and Computer Engineering,[**]
University of California,
Santa Barbara, CA 93106
††Mayachitra Inc.,
5266 Hollister Avenue,
Santa Barbara, CA 93111
solanki@mayachitra.com,anindya@ece.ucsb.edu,manj@ece.ucsb.edu

Abstract. A new, simple, approach for *active* steganography is proposed in this paper that can successfully resist recent blind steganalysis methods, in addition to surviving distortion constrained attacks. We present *Y*et *A*nother *S*teganographic *S*cheme (YASS), a method based on embedding data in randomized locations so as to disable the self-calibration process (such as, by cropping a few pixel rows and/or columns to estimate the cover image features) popularly used by blind steganalysis schemes. The errors induced in the embedded data due to the fact that the stego signal must be *advertised* in a specific format such as JPEG, are dealt with by the use of erasure and error correcting codes. For the presented JPEG steganograhic scheme, it is shown that the detection rates of recent blind steganalysis schemes are close to random guessing, thus confirming the practical applicability of the proposed technique. We also note that the presented steganography framework, of hiding in randomized locations and using a coding framework to deal with errors, is quite simple yet very generalizable.

Key words: data hiding, error correcting codes, steganalysis, steganography, supervised learning.

1 Introduction

Secure communication of a secret message has always been important to people, and it is not surprising that *steganography*, the art of communicating without revealing its existence, as well as *cryptography*, the art of concealing the meaning of a message, have a rich history. In this paper, we consider the problem of secure steganography via hiding information in digital images. In steganography, a *message* signal is embedded into a *host* or *cover* signal to get a *composite* or *stego* signal in such a way that the presence of hidden information cannot be detected by either *statistical* or *perceptual* analysis of the stego signal. In case of

[**] This research is supported in part by a grant from ONR # N00014-05-1-0816.

active steganography, there is an additional requirement that the hidden data must be recoverable even after benign or malicious processing of the stego signal by an adversary.

JPEG is arguably the most popular format for storing, presenting, and exchanging images. It is not surprising that steganography in the JPEG format, and its converse problem of steganalysis of JPEG images to find ones with hidden data, have received considerable attention from researchers over the past decade. There are many approaches and software available for JPEG steganography, which include OutGuess [1], StegHide [2], model-based steganography [3], perturbed quantization [4], F5 [5], and statistical restoration [6, 7].

Approaches for JPEG steganography have focused on hiding data in the least significant bit (LSB) of the quantized discrete cosine transform (DCT) coefficients. In order to avoid inducing significant perceptual distortion in the image, most methods avoid hiding in DCT coefficients whose value is 0. To detect the presence of data embedded in this manner, steganalysis algorithms exploit the fact that the DCT coefficient histogram gets modified when hiding random information bits. Hence recently proposed steganographic approaches attempt to match, as closely as possible, the original DCT histogram or its model. Westfield's F5 algorithm [5] increases, decreases, or keeps unchanged, the coefficient value based on the data bit to be hidden, so as to better match the host statistics. Provos's OutGuess [1] was the first attempt at explicitly matching the DCT histogram. Sallee proposed a model based approach for steganography [3] wherein the DCT coefficients were modified to hide data such that they follow an underlying model. Fridrich et al's perturbed quantization [4] attempts to resemble the statistics of a double-compressed image. Statistical restoration method proposed by Solanki et al [6, 7] can match the DCT histograms exactly, thus providing provable security so long as only the marginal statistics are used by the steganalyst.

Many steganalysis schemes (see [8–10]) have been able to successfully detect the above steganographic techniques that match marginal statistics or models. They exploit the fact that higher order statistics get modified by data hiding using these stego methods. It is known that, the higher order statistics, in general, are difficult to match, model, or restore. Recently, blind steganalysis algorithms [9–16] have been proposed that employ supervised learning to distinguish between the plain cover and stego images, and also identify the particular hiding algorithm used for steganography. These techniques bank on the fact that there are some image *features* that are modified during the embedding process which can be used as an input to the learning machine. For the success of this approach, it is crucial that these features are very sensitive to the embedding changes, but insensitive to the image content. This requires a good model for natural images against which the suspected stego images can be evaluated.

In spite of the absence of good universal models, recent steganalysis algorithms have been very successful by using a *self-calibration* method to approximate the statistics of the original cover (see, for example, Pevny and Fridrich [9, 10], and Dabeer et al [17]). The calibration method typically used for JPEG

steganography is quite simple; a few pixel rows and/or columns are cropped from the image so as to desynchronize it from the original JPEG grid and the resulting image is compressed again, which forms a good approximation of the cover image. The results reported in [10], the most recent multi-class JPEG steganalysis method that employs such self-calibration, are close to perfect: the steganalyst can determine one out of 6 stego algorithms employed for hiding with a detection accuracy of more than 95% in most cases, even at low embedding rates.

We present *yet another steganographic scheme* (YASS), a method for secure, active, steganography that can successfully resist the aforementioned blind steganalysis schemes. The technique is based on a simple idea of embedding data in random locations within an image, which makes it difficult for the steganalyst to get a good estimate of the cover image features via the self-calibration process. This approach works by desynchronizing the steganalyst, which is similar in spirit to what *stirmark* does for watermarks. Although data is hidden in randomly chosen blocks in the image, the image must be advertised in JPEG format. This leads to errors in the recovered data bits, which are dealt with by using erasure and error correcting codes in a manner similar to [18].

We evaluate the method against several recent blind steganalysis schemes, which include Farid's 72-dimensional wavelet-based features [19], Pevny and Fridrich's 23-dimensional DCT-based features [9] and 274-dimensional merged Markov and DCT features [10], a feature vector comprising of histogram of DCT coefficients, Xuan et al's statistical moments based spatial domain steganalysis features [13], and Chen et al's 324-dimensional JPEG steganalysis feature [20]. We find that the presented method, YASS, is completely undetectable for most of the embedding configurations, while the competing algorithms, OutGuess and StegHide, are detectable at the same hiding rates. We also note here that, because we use error correction coding framework, our method provides robustness against distortion constrained attacks thus enabling active steganography.

The rest of the paper is organized as follows. In Section 2, we motivate the use of the proposed randomized hiding for steganography. Next, in Section 3, we introduce and describe our JPEG steganographic technique: YASS. Experimental setup and results are presented in Section 4, followed by the concluding remarks in Section 5.

2 Resisting Blind Steganalysis

The notion of ϵ-security proposed by Cachin [21] states that a steganographic scheme is ϵ-secure if the Kullback-Leibler divergence between the cover and the stego signal distributions is less than a small number ϵ. This definition inherently assumes that cover signals can be described by "natural" distributions, which are known to the steganalyst. Statistical steganalysis schemes work by evaluating a suspected stego signal against an assumed or computed cover distribution or *model*. Most recent steganalysis schemes fall in this category [22, 10, 9].

Blind statistical steganalysis schemes use a supervised learning technique on *features* derived from plain cover as well as stego signals. This class of meth-

ods has been very successful in detecting steganographic methods available today. For example, detection results presented in [10] and also our own experiments indicate that popular JPEG steganographic schemes such as OutGuess [1], StegHide [2], model-based steganography [3], and 1D statistical restoration schemes [6, 7] can be successfully detected. Following are the key ingredients that contribute to the success of these blind steganalysis schemes.

1. **Self-calibration mechanism**: Calibration process is used by the blind steganalysis schemes to estimate the statistics of the cover image from the stego image. For JPEG steganography, this is typically achieved by decompressing the stego image to the spatial domain followed by cropping the image by a few pixels on each side and compressing the image again using the same compression parameters.
2. **Features capturing cover memory**: Most steganographic schemes hide data on a per-symbol basis, and typically do not explicitly compensate or preserve statistical dependencies. Hence, features that capture higher dimensional dependencies in the cover symbols are crucial in detecting the embedding changes. Cover memory has been shown to be very important to steganalysis [22], and is incorporated into the feature vector in several ways, e.g. [23, 15].
3. **Powerful machine learning**: Use of powerful machine learning techniques and training with several thousand images ensures that even the slightest statistical variation in the features is *learned* by the machine.

The calibration process is perhaps the most important of the above that allows the steganalyst to get an accurate underlying model of the cover image despite not having access to it. With this approach, statistical steganalysis is successful eventhough good *universal* statistical models for images are not available. In the following section, we discuss a simple approach that can potentially defeat these steganalysis schemes.

2.1 The Proposed Approach for Steganography

In order to enable secure communication in the presence of blind steganalysis, the steganographer must embed information into host signals in such a way that no image features are significantly perturbed during the embedding process. However, we must not forget that the steganalyst must depend on the stego image to derive the approximate cover image statistics via some sort of self-calibration process. The steganographer can, instead of (or along with) trying to preserve the feature vectors, embed data in such a way that it distorts the steganalyst's estimate of the cover image statistics. This can practically be achieved using the following approaches.

1. **Hiding with high embedding strength**: By embedding data with high strength, the cover image is distorted so much that the cover image statistics can no longer be derived reliably from the available stego image. This is indeed found to be true and reported in recent work by Kharrazi et al [24].

2. **Randomized hiding**: By randomizing the embedding approach, the algorithm to estimate the cover statistics can be effectively disabled. Things that can be randomized include the spatial location of hiding, the transform coefficient to hide, the choice of transform domain, or even the embedding method. In this manner, the steganalyst cannot make any consistent assumptions about the hiding process even if the embedding algorithm is known to everyone as per the Kerckhoff's principle.

There are some obvious disadvantages of using the first approach of hiding with high strength. First, the likelihood of perceptual distortion is high. Second, the data can possibly be detected by a steganalyst evaluating the stego image against a *universal* image model even if it is not that precise.

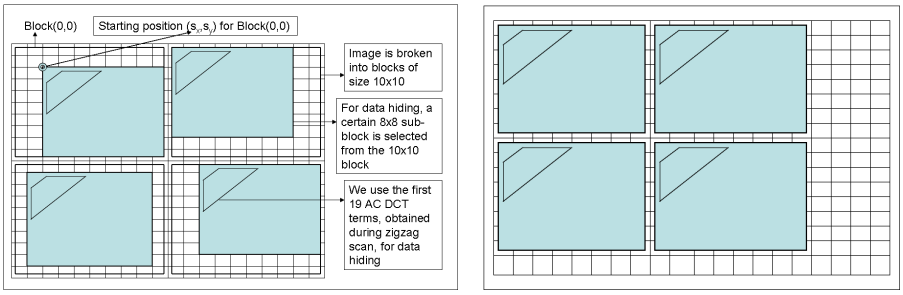
The second approach of hiding in a randomized manner is quite appealing, and we explore its simplest realization in this paper: embedding data in randomized locations within an image. One issue with hiding data in random locations is the possibility of encountering errors in the hidden bits due to the fact that the stego image must be *shipped* or *advertised* in a standard format such as JPEG. This is dealt with by the use of erasures and error correction coding framework previously employed in [18]. In the next section, we describe *y*et *a*nother *s*teganographic *s*cheme (YASS), a JPEG stegosystem based on the aforementioned framework.

3 YASS for JPEG Steganography

We now present a JPEG steganography scheme, YASS, that embeds data in 8×8 blocks whose locations are chosen randomly so that they do not coincide with the 8×8 grid used during JPEG compression. Let the host image be denoted by a $M \times N$ matrix of pixel values. For simplicity, we assume that the image is grayscale (single channel); if it is not, we extract its luminance. Below we describe the main steps involved in this randomized block hiding method.

1. Divide the image into blocks of size $B \times B$, where B, which we call *big block size*, is always greater than 8, the size of a JPEG block. Thus we have $M_B \times N_B$ big blocks in the image where $M_B = \lfloor \frac{M}{B} \rfloor$ and $N_B = \lfloor \frac{N}{B} \rfloor$.
2. For each block (i, j) $(0 \leq i < M_B, 0 \leq j < N_B)$, we pseudorandomly select a 8×8 sub-block in which to hide data. The key for the random number generator is shared between the encoder and the decoder. The pseudorandom number generator determines the location of the smaller 8×8 block within the big block. This process is illustrated in Figure 1(a) where four example blocks are shown, whose top leftmost corner (s_x, s_y) is randomly chosen from the set $\{0, 1, ..., B - 8\}$. Figure 1(b) shows the blocks as seen by the steganalyst who gets *out-of-sync* from the embedding blocks, and cannot resynchronize even if the embedding mechanism is known.
3. For every 8×8 block thus chosen, we compute its 2D DCT and divide it by a JPEG quantization matrix at a *design* quality factor QF_h. Data is hidden in a predetermined band of low frequency AC coefficients using quantization

index modulation. For maintaining perceptual transparency, we do not hide in coefficients that quantize to zero by the JPEG quantizer (following the selective embedding in coefficients scheme proposed in [18]).



(a) YASS hiding methodology: Data is hidden in randomly chosen 8×8 blocks within a big block of size $B \times B$, with $B > 8$. This example uses $B = 10$.

(b) The block structure as seen by a steganalyst, who gets *out-of-sync* with the blocks used during embedding, and cannot synchronize even if the hiding method and its parameters are known.

Fig. 1. The embedding method and its detection.

Note that using this approach, we can effectively de-synchronize the steganalyst so that the features computed by him would not directly capture the modifications done to the image for data hiding (see Figure 1). It should be noted that with this embedding procedure, we are reducing the embedding rate in two ways. First, some *real estate* of the image is wasted by choosing bigger blocks from which an 8×8 block is chosen to hide data. Note that the above framework can be further generalized to enable lesser *wastage*, by using larger big blocks and putting more 8×8 blocks into them. For example, we can use big blocks of size 33×33 and embed in sixteen 8×8 blocks within. We report results for such implementations as well (Section 4). The second cause of decrease in rate is that since the embedding grid does not coincide with the JPEG grid, there are errors in the received data which must be corrected by adding redundancy. This process is briefly described in the following section.

3.1 Coding Framework

In order to deal with the errors caused in the image due to JPEG compression, we use a coding framework using repeat-accumulate (RA) codes [25], similar to that proposed in [18]. This framework also allows us to hide in an adaptive fashion, avoiding coefficients that quantize to zero so as to control the perceptual distortion to the image.

For every block, we consider an embedding band comprising of first n low frequency coefficients which forms the *candidate embedding band*. Data bits are hidden in a coefficient lying in the band if it does not quantize to zero using the JPEG quantizer at QF_h. Before the hiding process, the bit stream to be hidden

is coded, using a low rate code, assuming that all host coefficients that lie in the candidate embedding band will actually be employed for hiding. A code symbol is *erased at the encoder* if the local adaptive criterion (of being quantized to zero) for the coefficient is not met. A rate 1/q RA encoder is employed, which involves q-fold repetition, pseudorandom interleaving and accumulation of the resultant bit-stream. Decoding is performed iteratively using the sum-product algorithm [26]. The use of this coding framework for YASS provides the following advantages.

1. **Protection against initial JPEG compression**: Use of the coding framework provides error-free recovery of the hidden data after the initial JPEG compression so that the image can be advertised in the JPEG format.
2. **Flexibility in choosing hiding locations**: The coding framework allows us to dynamically select the embedding locations in order to limit the perceptual distortion caused to the host image during hiding. It is well known that embedding in DCT coefficients that quantize to zero can lead to visible artifacts in the stego image.
3. **Enabling active steganography**: The use of error correcting codes also provides protection against several distortion constrained attacks that an active warden might perform. The attacks that can be survived include a second JPEG compression, additive noise, limited amount of filtering, and so on. This provides a significant advantage over most other stego methods available in the literature.

4 Experiments and Results

We conduct a comprehensive set of experiments and present the results demonstrating the applicability of the presented approach. First, the results for the embedding capacity are presented for some standard images (Section 4.1). Next, in Section 4.2, we present the detection results of our scheme using recent blind steganalysis methods for a couple of datasets (comprising of several thousand natural images). We also compare the detection results of our method with those of OutGuess and StegHide in Section 4.3, and show that our methods clearly outperform these approaches. As a control experiment, in Section 4.4, we compare our hiding method with a *naive* standard hiding approach, in which data is hidden in randomly chosen low-frequency DCT coefficients using standard JPEG grid, while keeping the same embedding rate.

4.1 Embedding Rates

In Table 1, we list the number of bits that can be hidden in several standard images using YASS with different embedding parameters. QF_h denotes the design quality factor used during hiding, and QF_a denotes the output or advertised image quality factor. Note that for the presented scheme, these two can be different and their values do affect the steganalysis performance (see Section 4.2).

The same notations are used in other tables presented in this paper. Also note that the number of bits reported in this section are before coding and can be recovered without any errors, even in the presence of distortion constrained attacks. For all the results presented in this paper, we use 19 low-frequency DCT coefficients as the candidate embedding band.

Table 1. Number of information bits that can be hidden in some standard images of size 512×512. The big-block size $B = 9$.

$QF_h \rightarrow QF_a$	baboon	peppers	airplane	lena	couple	crowd
50 → 50	1453	1295	2128	1702	1655	2979
50 → 75	14896	6620	7448	7448	9930	11916
75 → 75	1453	1805	2590	2128	2128	3972
60 → 75	11916	6620	7448	6620	8512	9930

From this table, we see that the number of bits that can be embedded increases when $QF_a > QF_h$, however this also leads to slightly more successful steganalysis performance (to be shown in Tables 4 and 6), which gives us a trade-off between the embedding rate and the detection performance. When QF_a equals QF_h, there are more physical errors in the channel leading to a reduced embedding rate.

In Table 2, we study the effect of using different big-block sizes B on the embedding capacity. Here we also report the bits per non-zero coefficients (*bpnc*). It is seen from this table that using lower B provides higher embedding capacity, which is because we end-up using more *real estate* of the host image for hiding. We can ensure even higher percentage of image utilization by using larger big-blocks and putting greater number of 8×8 blocks within. For example, we can put four 8×8 blocks in a 17×17 block, thus making an effective big-block size of $B_{eff} = \frac{17}{2} = 8.5$. We experiment with block-size $B = (n{\times}8 + 1)$ and use n^2 8×8 blocks for hiding, and report the results in Table 3. The embedding rate improves as we use more image area for hiding, but the detection results get worse (from the steganographer's viewpoint, see Tables 4 and 6). Also, using a lower B_{eff} does not always guarantee a higher embedding capacity because of the fact that more errors may be induced due to JPEG compression, which forces us to use a larger redundancy factor, q, of the RA code.

4.2 Detection Results

The steganographic security of our scheme is evaluated against the following blind steganalysis schemes. The names in **bold** are the ones used to denote the steganalysis schemes in the tables presented in this paper.

1. **Farid**: 72-dimensional feature vector based on moments in the wavelet domain [19].
2. **PF-23**: Pevny and Fridrich's 23-dimensional DCT feature vector [9].
3. **PF-274**: Pevny and Fridrich's 274-dimensional feature vector that merges Markov and DCT features [10].

Table 2. Hiding rates for the 512×512 Lena image for different big-block sizes B. *bpnc* denotes bits per non-zero coefficients and *databits* denotes the number of hidden information bits.

$QF_h \rightarrow QF_a$	B	9	B	10	B	12	B	14
50 → 50	databits	1702	databits	1497	databits	882	databits	464
	bpnc	0.072	bpnc	0.064	bpnc	0.038	bpnc	0.020
50 → 75	databits	7448	databits	5491	databits	4189	databits	2736
	bpnc	0.213	bpnc	0.159	bpnc	0.118	bpnc	0.077
75 → 75	databits	2128	databits	2059	databits	1457	databits	794
	bpnc	0.059	bpnc	0.057	bpnc	0.040	bpnc	0.022
60 → 75	databits	6620	databits	5491	databits	3724	databits	2462
	bpnc	0.187	bpnc	0.158	bpnc	0.105	bpnc	0.069

Table 3. Hiding rates for the 512×512 Lena image for larger big-block sizes $B = 9$, 25, 49, 65 and 81, which can incorporate several 8×8 blocks.

$QF_h \rightarrow QF_a$	B	9	B	25	B	49	B	65	B	81
50 → 75	databits	7448	databits	7834	databits	8064	databits	7963	databits	8064
	bpnc	0.213	bpnc	0.224	bpnc	0.235	bpnc	0.228	bpnc	0.229
75 → 75	databits	2128	databits	2350	databits	2341	databits	1837	databits	1577
	bpnc	0.059	bpnc	0.065	bpnc	0.065	bpnc	0.051	bpnc	0.044
60 → 75	databits	6620	databits	7050	databits	8064	databits	7166	databits	7258
	bpnc	0.187	bpnc	0.200	bpnc	0.231	bpnc	0.203	bpnc	0.205

4. **DCT hist.**: Histogram of DCT coefficients from a low-frequency band [7].
5. **Xuan-39**: Spatial domain steganalysis proposed by Xuan et al [13] (a 39-dimensional feature vector).
6. **Chen-324**: JPEG steganalysis based on statistical moments of wavelet characteristic functions proposed by Chen et al [20].

Since we decompress the images at the time of hiding, it can be argued that spatial domain steganalysis schemes may be able to detect the presence of embedded data. Hence we include a recent spatial domain steganalysis scheme in our tests.

We conduct our experiments on two datasets: the first having 4500 images in compressed (JPEG) format and the other having 2000 images in uncompressed TIFF format. For each dataset, we use half the data for training and the other half for testing. The training and testing sets have an equal number of cover and stego images. The idea behind using datasets in different formats is to study the effect of using already compressed images verses uncompressed images for our method since it hides in a desynchronized 8×8 grid. As we see later, there is not much difference in the observed detection rates for the two datasets. As in all published blind steganalysis approaches, we train a support vector machine (SVM) on a set of known stego and cover images and use the threshold thus obtained, to distinguish between the cover and stego images at the time of testing.

The SVM classifier has to distinguish between two class of images: cover (class '0') and stego (class '1'). Let X_0 and X_1 denote the events that the actual image being observed belongs to classes '0' and '1', respectively. On the detection side, let Y_0 and Y_1 denote the events that the observed image is classified as belonging to classes '0' and '1', respectively. We use the probability of detection, P_{detect} as our evaluation criteria, which is defined as follows.

$$\begin{aligned} P_{detect} &= 1 - P_{error} \\ P_{error} &= P(X_0)P(Y_1|X_0) + P(X_1)P(Y_0|X_1) \\ &= \frac{1}{2}P_{FA} + \frac{1}{2}P_{miss}, \text{ for } P(X_0) = P(X_1) = \frac{1}{2} \end{aligned}$$

where $P_{FA} = P(Y_1|X_0)$ and $P_{miss} = P(Y_0|X_1)$ denote the probability of false alarm and missed detection respectively. Note that the above equation assumes an equal number of cover and stego images in the dataset. For the steganalysis results, we report P_{detect} upto 2 significant digits after the decimal point. An uninformed detector can classify all the test images as stego (or cover) and get an accuracy of 0.5. Thus, P_{detect} being close to 0.5 implies nearly undetectable hiding, and as the detectability improves, P_{detect} should increase towards 1.

Table 4. JPEG dataset: Steganalysis results for randomized block based hiding, when big-block size B is varied. It can be seen that the detection is random for most of the configurations.

QF_h (used for hiding)	QF_a (advertised QF)	Steganalysis Method	Detection accuracy: P_{detect}			
			B=9	B=10	B=12	B=14
50	50	Farid	0.52	0.51	0.52	0.51
50	75	Farid	0.55	0.55	0.54	0.51
75	75	Farid	0.52	0.51	0.52	0.51
50	50	PF-23	0.56	0.55	0.54	0.54
50	75	PF-23	0.59	0.59	0.56	0.60
75	75	PF-23	0.53	0.57	0.53	0.52
50	50	PF-274	0.58	0.56	0.53	0.55
50	75	PF-274	0.77	0.79	0.74	0.65
75	75	PF-274	0.59	0.60	0.62	0.54
50	50	DCT-hist	0.53	0.53	0.51	0.53
50	75	DCT-hist	0.64	0.64	0.60	0.54
75	75	DCT-hist	0.55	0.54	0.55	0.53
50	50	Xuan-39	0.54	0.56	0.54	0.51
50	75	Xuan-39	0.63	0.64	0.57	0.53
75	75	Xuan-39	0.52	0.54	0.53	0.52
50	50	Chen-324	0.57	0.51	0.55	0.54
50	75	Chen-324	0.75	0.65	0.60	0.55
75	75	Chen-324	0.54	0.55	0.53	0.53

In Tables 4 and 5, we present the detection accuracy obtained on the JPEG and TIFF image dataset respectively. Note that even for the TIFF dataset, the

Table 5. TIFF dataset: Steganalysis results for randomized block based hiding, when the big-block size B is varied. It can be seen that the detection is random for most of the configurations.

QF_h (used for hiding)	QF_a (advertised QF)	Steganalysis Method	Detection accuracy: P_{detect}			
			B=9	B=10	B=12	B=14
50	50	Farid	0.51	0.51	0.51	0.51
50	75	Farid	0.53	0.51	0.52	0.51
75	75	Farid	0.50	0.51	0.51	0.51
50	50	PF-23	0.53	0.55	0.53	0.53
50	75	PF-23	0.59	0.66	0.53	0.53
75	75	PF-23	0.54	0.51	0.53	0.53
50	50	PF-274	0.52	0.56	0.55	0.51
50	75	PF-274	0.72	0.81	0.65	0.60
75	75	PF-274	0.56	0.56	0.52	0.53
50	50	DCT-hist	0.51	0.53	0.52	0.48
50	75	DCT-hist	0.60	0.59	0.56	0.56
75	75	DCT-hist	0.52	0.52	0.51	0.52
50	50	Xuan-39	0.52	0.51	0.52	0.48
50	75	Xuan-39	0.57	0.60	0.53	0.48
75	75	Xuan-39	0.47	0.51	0.49	0.47
50	50	Chen-324	0.55	0.55	0.53	0.53
50	75	Chen-324	0.65	0.64	0.56	0.54
75	75	Chen-324	0.55	0.51	0.51	0.50

output is always a JPEG image at an advertised quality factor. For this dataset, the plain cover images are JPEG compressed at QF_a during the training as well as testing. It can be seen from the tables that our scheme is undetectable using any of the steganalysis features. The only time the detection is not completely random is when the design quality factor is lower than the advertised one ($QF_h = 50$ and $QF_a = 75$). We also present, in Table 6, the steganalysis results when larger big-block sizes are used, so as to incorporate more 8×8 blocks within. P_{detect} remains close to 0.5 in most cases, but for PF-274, it increases as B increases (in Table 6), since more area of the image is used for hiding when employing larger big-blocks. Note that since we are using a set of images and the embedding rate varies for individual images, we cannot provide *bpnc* values in these tables.

4.3 Comparison with Competing Methods

In Table 7, we present a comparison of our steganographic scheme, YASS, to OutGuess[1] and StegHide[2]. To enable fair comparison, we must use the same hiding rate for all the schemes. This is complicated by the fact that YASS uses an error correcting code whose redundancy factor, q, determines the actual number of information bits hidden in the image. Experiments indicate that, for images in our dataset, the redundancy factor required to ensure zero BER was in the range 10-40, which depends on the particular image and the level of JPEG compression

Table 6. Using larger big-block size B: Steganalysis results for randomized block based hiding on the TIFF image dataset, for block-size $B = 9$, 25 and 49. For 9×9, 25×25 and 49×49 blocks, we use 1, 9 and 36 8×8 sub-blocks respectively for hiding.

QF_h (used for hiding)	QF_a (advertised QF)	Steganalysis Method	Detection accuracy: P_{detect}		
			B=9	B=25	B=49
50	50	Farid	0.51	0.50	0.50
50	75	Farid	0.53	0.51	0.52
75	75	Farid	0.50	0.49	0.51
50	50	PF-23	0.53	0.53	0.57
50	75	PF-23	0.59	0.58	0.67
75	75	PF-23	0.54	0.53	0.53
50	50	PF-274	0.52	0.57	0.59
50	75	PF-274	0.72	0.73	0.78
75	75	PF-274	0.56	0.58	0.68
50	50	DCT-hist	0.51	0.50	0.51
50	75	DCT-hist	0.60	0.56	0.50
75	75	DCT-hist	0.52	0.51	0.50
50	50	Xuan-39	0.52	0.52	0.52
50	75	Xuan-39	0.57	0.53	0.50
75	75	Xuan-39	0.47	0.52	0.51
50	50	Chen-324	0.55	0.53	0.52
50	75	Chen-324	0.65	0.58	0.57
75	75	Chen-324	0.55	0.52	0.51

involved. Hence we present results where the amount of data embedded using OutGuess and StegHide corresponds to the hiding rate obtained using $q = 10$ and $q = 40$. It can be seen from the table that both OutGuess and StegHide are almost completely detectable (especially when using PF-23, PF-274, and DCT-hist features), but YASS is not detectable at equivalent hiding rates.

We also experimented with the F5 steganographic scheme [5], which uses *matrix embedding* (see for example [27]), and found that this scheme is also undetectable at equivalent embedding rates using PF-274 and PF-23 features. Matrix embedding allows hiding at *embedding efficiencies* (defined as number of bits hidden for every change made to the host symbols), potentially much higher than the trivial efficiency of 2 bits per change (bpc). Thus for passive steganography, schemes that employ matrix embedding (such as F5) can enable undetectable hiding at embedding rates equivalent to YASS. However, an added advantage of YASS is that it can provide protection against active adversaries by its use of error correcting codes.

4.4 Comparison with Standard Hiding at Same Rate

In the previous sections, it is seen that our proposed steganographic approach performs quite well against blind steganalysis schemes that we have tested against. We must, however, note that the improved steganographic security comes at the

Table 7. Steganalysis results for comparing the randomized block based scheme (YASS) with OutGuess and Steghide schemes, used at rates of $\frac{1}{10}$ and $\frac{1}{40}$, for the TIFF image dataset. For OutGuess and Steghide, the images are JPEG compressed using a quality factor of QF_a before being presented to the steganographic scheme. Note that the QF_h parameter is applicable only for the YASS scheme.

QF_h	QF_a	Steganalysis Method	Detection accuracy: P_{detect}				
			YASS	OutGuess-$\frac{1}{10}$	Steghide-$\frac{1}{10}$	OutGuess-$\frac{1}{40}$	Steghide-$\frac{1}{40}$
50	50	Farid	0.51	0.74	0.50	0.77	0.52
50	75	Farid	0.53	0.59	0.50	0.50	0.55
75	75	Farid	0.50	0.59	0.50	0.50	0.55
50	50	PF-23	0.53	0.98	0.78	0.97	0.80
50	75	PF-23	0.59	1.00	0.99	0.99	0.99
75	75	PF-23	0.54	1.00	0.99	0.99	0.99
50	50	PF-274	0.52	1.00	0.98	1.00	0.96
50	75	PF-274	0.72	1.00	1.00	1.00	1.00
75	75	PF-274	0.56	1.00	1.00	1.00	1.00
50	50	DCT-hist	0.51	0.95	0.59	0.94	0.60
50	75	DCT-hist	0.60	1.00	0.91	1.00	0.93
75	75	DCT-hist	0.52	1.00	0.91	1.00	0.93

cost of reduced embedding rate. To further investigate whether the good performance of YASS is simply because of reduced rate or not, we present another *simpler* extension of the idea of embedding in random locations: we now embed data in randomly chosen low-frequency AC DCT coefficients computed using the original JPEG grid. This approach would incur minimal distortion to the original cover during the hiding process, and hence would be an ideal *control* experiment for testing our scheme. The results are reported in Table 8, where we compare the YASS embedding scheme with this randomized frequency (RF) scheme for three hiding rates: 2 out of 19, 1 out of 19, and 1 out of 38 coefficients. It can be seen that the naive RF scheme performs quite well, however, the performance of YASS is consistently better.

We end this section by noting that the reported results (for RF) are for embedding at trivial embedding efficiency of 2 bpc. As stated earlier, the detectability can be reduced by causing fewer changes to the DCT coefficients by using matrix embedding. One of the key factors that determines detectability is the change rate. This change rate can be thought of as the encoders' "budget", which can be "used" either way: to provide robustness by using redundancy and giving up some embedding efficiency, or to improve the embedding efficiency via matrix embedding but causing an increase in the fragility of the system. YASS goes the first way: it can provide robustness against an active adversary by using error correcting codes but the embedding efficiency is quite low. However, we emphasize that for passive warden steganography, an equivalent level of undetectability can be achieved at the same embedding rates using matrix embedding.

Table 8. Comparison of steganalysis results for randomized block (RB) hiding (i.e., YASS) with $B = 9$, and randomized frequency (RF) based hiding.

QF_h (used for hiding)	QF_a (advertised QF)	Steganalysis Method	Detection accuracy: P_{detect}			
			RB (YASS)	RF 1/38	RF 1/19	RF 2/19
50	50	Farid	0.51	0.66	0.65	0.67
75	75	Farid	0.50	0.52	0.58	0.67
50	50	PF-23	0.53	0.69	0.83	0.92
75	75	PF-23	0.54	0.58	0.72	0.83
50	50	PF-274	0.52	0.71	0.79	0.90
75	75	PF-274	0.56	0.62	0.75	0.82
50	50	DCT-hist	0.51	0.55	0.68	0.86
75	75	DCT-hist	0.52	0.54	0.59	0.80

5 Conclusions

In this paper, we have demonstrated a simple yet very effective steganographic approach that provides security against recent blind steganalysis schemes. The method embeds data in randomly chosen host blocks, thus relying on confusing the steganalyst's estimate of the cover statistics, rather than preserving the host image features. We note that the improved security comes at the cost of embedding capacity, and in the future, we will investigate schemes that can significantly increase the hiding capacity while maintaining the steganographic security. We will also explore other mechanisms for randomizing the embedding process, such as using randomly chosen transform domains.

References

1. Provos, N.: Defending against statistical steganalysis. In: 10th USENIX Security Symposium, Washington DC, USA (2001)
2. Hetzl, S., Mutzel, P.: A graph theoretic approach to steganography. In: 9th IFIP TC-6 TC-11 International Conference, Communications and Multimedia Security. Volume 3677., Salzburg, Austria (2005) 119–128
3. Sallee, P.: Model-based steganography. In: IWDW 2003, LNCS 2939. (2003) 154–167
4. Fridrich, J., Goljan, M., Lisoněk, P., Soukal, D.: Writing on wet paper. In: ACM Workshop on Multimedia and security, Magdeburg, Germany (2004)
5. Westfeld, A.: High capacity despite better steganalysis (F5 - a steganographic algorithm). In: Lecture notes in computer science: 4th International Workshop on Information Hiding. Volume 2137. (2001) 289–302
6. Solanki, K., Sullivan, K., Madhow, U., Manjunath, B.S., Chandrasekaran, S.: Statistical restoration for robust and secure steganography. In: Proc. ICIP. (2005) II–1118–21
7. Solanki, K., Sullivan, K., Madhow, U., Manjunath, B.S., Chandrasekaran, S.: Probably secure steganography: Achieving zero K-L divergence using statistical restoration. In: Proc. ICIP. (2006) 125–128
8. Wang, Y., Moulin, P.: Steganalysis of block-DCT image steganography. In: IEEE workshop on Statistical Signal Processing, St Louis, MO, USA (2003)

9. Pevny, T., Fridrich, J.: Multi-class blind steganalysis for JPEG images. In: Proc. of SPIE, San Jose, CA (2006)
10. Pevny, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. In: Proc. of SPIE, San Jose, CA (2007)
11. Avcibas, I., Sankur, B., Memon, N.: Image steganalysis with binary similarity measures. In: Proc. ICIP. (2002) 645–648
12. Lyu, S., Farid, H.: Detecting hidden messages using higher-order statistics and support vector machines. In: Lecture notes in computer science: 5th International Workshop on Information Hiding. Volume 2578. (2002)
13. Xuan, G., Shi, Y.Q., Gao, J., Zou, D., Yang, C., Yang, C., Zhang, Z., Chai, P., Chen, C., Chen, W.: Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions. In: Lecture notes in computer science: 7th International Workshop on Information Hiding. (2005)
14. Harmsen, J.J., Pearlman, W.A.: Steganalysis of additive noise modelable information hiding. In: Proc. of SPIE. (2003) 131–142
15. Shi, Y.Q., Chen, C., Chen, W.: A Markov process based approach to effective attacking JPEG steganography. In: Lecture notes in computer science: 8th International Workshop on Information Hiding. (2006)
16. Wang, Y., Moulin, P.: Optimized feature extraction for learning-based image steganalysis. IEEE Transactions on Information Forensics and Security **2**(1) (2007) 31–45
17. Dabeer, O., Sullivan, K., Madhow, U., Chandrasekaran, S., Manjunath, B.: Detection of hiding in the least significant bit. IEEE Transactions on Signal Processing, Supplement on Secure Media I **52**(10) (2004) 3046–3058
18. Solanki, K., Jacobsen, N., Madhow, U., Manjunath, B.S., Chandrasekaran, S.: Robust image-adaptive data hiding based on erasure and error correction. IEEE Trans. on Image Processing **13**(12) (2004) 1627 –1639
19. Farid, H.: http://www.cs.dartmouth.edu/farid/research/steg.m. (Code for generating wavelet-based feature vectors for steganalysis.)
20. Chen, C., Shi, Y.Q., Chen, W., Xuan, G.: Statistical moments based universal steganalysis using JPEG-2D array and 2-D characteristic function. In: Proc. ICIP, Atlanta, GA, USA (2006) 105–108
21. Cachin, C.: An information theoretic model for steganography. LNCS: 2nd Int'l Workshop on Info. Hiding **1525** (1998) 306–318
22. Sullivan, K., Madhow, U., Chandrasekaran, S., Manjunath, B.: Steganalysis for Markov cover data with applications to images. IEEE Transactions on Information Forensics and Security **1**(2) (2006) 275–287
23. Fu, D., Shi, Y.Q., Zou, D., Xuan, G.: JPEG steganalysis using empirical transition matrix in block dct domain. In: International Workshop on Multimedia Signal Processing, Victoria, BC, Canada (2006)
24. Kharrazi, M., Sencar, H.T., Memon, N.: Cover selection for steganographic embedding. In: Proc. ICIP. (2006) 117–120
25. Divsalar, D., Jin, H., McEliece, R.J.: Coding theorems for turbo-like codes. In: 36th Allerton Conf. on Communications, Control, and Computing. (1998) 201–210
26. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. on Info. Theory **47**(2) (2001) 498–519
27. Fridrich, J., Soukal, D.: Matrix embedding for large payloads. In: Proc. of SPIE. (2006) 727–738