Further Study on YASS: Steganography Based on Randomized Embedding to Resist Blind Steganalysis

Anindya Sarkar†, Kaushal Solanki†† 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

ABSTRACT

We present further extensions of *yet another steganographic scheme* (YASS), a method based on embedding data in randomized locations so as to resist blind steganalysis. YASS is a JPEG steganographic technique that hides data in the discrete cosing transform (DCT) coefficients of randomly chosen image blocks. Continuing to focus on JPEG image steganography, we present, in this paper, a further study on YASS with the goal of improving the rate of embedding. Following are the two main improvements presented in this paper: (i) a method that randomizes the quantization matrix used on the transform domain coefficients, and (ii) an iterative hiding method that utilizes the fact that the JPEG "attack" that causes errors in the hidden bits is actually known to the encoder. We show that using both these approaches, the embedding rate can be increased while maintaining the same level of undetectability (as the original YASS scheme). Moreover, for the same embedding rate, the proposed steganographic schemes are more undetectable than the popular matrix embedding based F5 scheme, using features proposed by Pevny and Fridrich for blind steganalysis.

Keywords: covert communication, data hiding, randomized hiding, steganalysis, steganography.

1. INTRODUCTION

Research in steganalysis, the science of detecting the presence of hidden message in an innocuous-looking host, has taken great strides in the past few years. Many popular steganographic methods (such as[1–4]) can now be detected using *blind* steganalysis schemes (such as[5–11]) that use powerful machine learning methods to train a classifier from known examples of hidden and plain cover signals. These schemes have been successful not only in detecting the presence of embedded data, but also in identifying the particular steganographic scheme used for hiding so that further analysis can be done to recover the embedded message. The success of these methods can be attributed to their use of *features vectors that incorporate higher-order joint statistics*, and their use of a *self-calibration mechanism*[5,6] to reliably estimate the cover signal statistics from the available stego signal.

Our recently proposed steganographic method called *yet another steganographic scheme* (YASS)[12] is arguably the first active steganographic scheme that has been shown to resist the aforementioned blind steganalysis schemes, albeit with a relatively low hiding capacity. The idea is to embed data in randomized locations so as to disable the self-calibration process used by the blind steganalysis schemes. YASS is a JPEG steganographic technique that hides data in the discrete cosing transform (DCT) coefficients of randomly chosen image blocks.

Continuing to focus on JPEG image steganography, we present, in this paper, a further study on YASS with the goal of improving the rate of embedding. Following are the two main improvements presented in this paper.

Further author information: (Send correspondence to Anindya Sarkar. E-mail: anindya@ece.ucsb.edu, Telephone: 1-805-893-5682

1. **Further randomization**: We present results for a *more randomized* scheme in which the quantization matrix used on the transform domain coefficients has been randomized. The design quality factor is varied among the different image blocks and its value, for a given block, is determined based on the local image variance.

2. **Attack-aware embedding**: These schemes utilize the fact that the embedded image undergoes JPEG compression before it is "advertised". As explained in,[12] this compression acts as an attack to the embedding system which causes errors and reduces the embedding rate. The good news is that this *attack* is known at the encoder and its effect can be reduced via an iterative embedding process.

The results obtained are quite encouraging. The rate of embedding, measured in bits per non-zero coefficients (bpnc), has been improved while maintaining the undetectability against recent blind steganalysis schemes. An important result is that the attack-aware embedding schemes outperform the F5 algorithm,[13] which uses matrix embedding, in terms of the observed detection rates when data is hidden at equivalent embedding rates. The merged DCT and Markov features proposed by Pevny and Fridrich[6] have been used in these tests, which have also been found to be among the most successful steganalysis schemes for detecting JPEG image steganography.

2. BACKGROUND: RESISTING BLIND STEGANALYSIS USING RANDOMIZED EMBEDDING

The notion of ϵ-security proposed by Cachin[14] 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*. *Blind* statistical steganalysis schemes use a supervised learning technique trained on *features* derived from plain cover as well as stego signals. This class of methods has been very successful in detecting steganographic methods available today. For example, detection results presented in[6] and also our own experiments indicate that popular JPEG steganographic schemes such as OutGuess,[2] StegHide,[3] model-based steganography,[1] and 1D statistical restoration schemes[4,15] can be successfully detected.

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. As explained in,[12] this can potentially be achieved either by embedding with a high strength, or by randomizing the hiding process. While the first approach of hiding with high strength has several disadvantages, such as possibility of high perceptual distortion and risk of detection via an universal image model, the second approach of randomized hiding is appealing and is employed in the present paper.

The specific implementation of the idea of randomized hiding for JPEG steganography is the YASS embedding scheme, which works by embedding data in 8×8 blocks whose locations are randomly chosen so that they are different from the regular 8×8 grid used for JPEG compression. Instead of hiding in regular 8×8 grids, a grid of bigger blocks (of size $B > 8$, where B is called the *big block size*) is formed from which an 8×8 block is chosen randomly to hide data. 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. 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 is the main cause of the relatively lower embedding rate of the YASS scheme.

3. IMPROVING THE EMBEDDING RATE

We now present two different approaches that we have studied with the goal of improving the embedding rate while maintaining the undetectability of the schemes against recent blind steganalysis. The first approach is a

natural extension of the YASS framework, in which, in addition to choosing randomized locations, we also vary the design quality factor to be used for hiding per block. In this manner, we expect to cause different statistical changes to different parts of an image, and a steganalysis scheme that computes statistics over the entire image is likely to get more confused (as compared to the prior scheme in which only the hiding location is randomized). It should be noted that the range of variation of the hiding parameters is adjusted such that the perceptual transparency of the stego image is maintained. The second approach, which provides better improvement in rate, utilizes the fact that the JPEG compression "attack" that causes errors in the embedded bitstream is known at the encoder. Note that both the approaches can in principle be combined into a third approach which can potentially provide further improvement in the rate. Below we discuss the two approaches in more detail.

3.1 Further randomization: A mixture based approach

As described in Section 2, the YASS method works by randomly selecting an 8×8 block in a $B \times B$ *big block* followed by embedding in a low-frequency band within the 8×8 block. In the original proposal,[12] the location of this block was the only random parameter. We here present a method that explores other avenues of randomization within the same overall framework. We select different quantization matrices for different blocks by choosing different design quality factors. Thus, a *mixture* of various parameters is used for hiding.

The mixture based approach works in the same way as the original YASS scheme with the only difference being that the design quality factor (and hence the quantization matrix) varies for every block. Thus, similar to YASS, 8×8 blocks are chosen randomly within a *big block* of size $B \times B$, and data is hidden by quantizing the non-zero coefficients in a low-frequency band, after diving by the quantization matrix, which, in this case, varies from block-to-block. We have explored the following means of varying the quantization matrices.

- *Random variation*: Here we choose the quality factor randomly from a predefined set. The secret key used for the choice of the quality factor as well as the predefined set of quality factors used during hiding are shared with the decoder. Pseudo random selection is the simplest and a natural way to vary the hiding parameter.
- *Image adaptive variation:* We also explore the use of a couple of image-adaptive methods for choosing the embedding parameters. Note that if a content-specific choice of the the hiding parameter is made, the information cannot be conveyed to the decoder. In this scenario, the decoder must determine the hiding parameter value based on the image content itself, which may vary due to the embedding process as well as any attacks. Thus the criteria used in determining the embedding parameter must be robust to compression attacks. Note that these schemes are similar in spirit to the entropy thresholding scheme.[16]
  - *Based on coefficient count:* the quality factor used for hiding is varied based on the count of the non-zero coefficients at a selected quality factor. Our experiments indicate that though this leads to greater number of embedded bits, the advantage is lost due to an increased number of instances when the decoder makes a mistake in guessing the right embedding parameter.
  - *Based on block variance:* the quality factor is varied based on the variance of the block. Unlike the coefficient count, the block variance was found to be a robust criterion for the choice of the hiding parameter. The value of the variance did not vary much, between the cover and stego images, in presence of attacks, and hence, the decoder made fewer mistakes in guessing the hiding parameters correctly. The variance values are divided into as many partitions as the number of different quality factors in the mixture based scheme. Note that the choice of the variance based partitions, where each partition corresponds to a different design quality factor, is highly non-uniform; most of the blocks have very low variance for natural images and only a small fraction has variance significantly greater than zero. The choice of these partitions is made experimentally so as to ensure that the overall embedding rate improves.

The use of erasures and errors correcting codes ensures that all the embedded bits can be decoded successfully in spite of the fact that the decoder will occasionally make mistakes in guessing the parameter values for the image adaptive scheme discussed above. The adaptive method based on block variance is the most promising of the above methods and we now discuss it more specifically below.

3.1.1 Varying the design quality factor based on local variance

For natural images, most blocks have essentially a higher low-frequency content. The distribution of the variance values among the different image blocks is essentially a tapered distribution: it is higher for the low variance values and there is a steady decrease with increasing variance values. For simplicity, let us consider a 3-mixture case with quality factors QF_1, QF_2 and QF_3, where $QF_1 < QF_2 < QF_3$. Let the variance be partitioned using $[0, x_1, x_2, \infty)$, where

- QF_3 is allocated to zone $[0, x_1)$,
- QF_2 is allocated to zone $[x_1, x_2)$,
- QF_1 is allocated to zone $[x_2, \infty)$.

Due to the high concentration of values near 0 and the steady decrease in the distribution with increase in variance, the number of blocks having variance in the range $[0, x_1)$ is maximum and the number of blocks with variance in the $[x_2, \infty)$ is minimum. When hiding is done at higher QF, it is generally more difficult to detect it, as shown later in Tables 1-2. Hence, the highest QF (QF_3) is allocated to the zone with maximum number of blocks, $[0, x_1)$. Also, for increasing the hiding capacity, the embedding rate is increased for a certain QF if the number of erasures is less for that QF (i.e. there are more coefficients available for hiding), and if the attack becomes less severe. If the quality factor to be used for attack is 75, then as the QF is increased, say from 20-75, the erasure rate progressively decreases. However, the effect of the attack becomes progressively more severe - e.g. an attack at QF=75 is more severe when the design QF=70 than when it is 20. Through experiments (Table 1), it is seen that the embedding rate is maximum at QF=50 and it decreases for both higher and lower QF values. Now, if the mixture used for hiding has QF values of 50, 60 and 70, we should use a QF of 50 for those blocks where there are a substantial number of non-zero terms available for hiding, i.e. zone $[x_2, \infty)$. For QF=70, the embedding rate at this design QF is quite low and so, we do not lose much, from an embedding rate perspective, by using QF=70 for blocks with variance values in $[0, x_1)$. In general, blocks with higher variance allow for more embedding than those with lower variance. Thus, the philosophy here is to maximize the hiding rate by using that design QF which results in higher rate in those blocks which allow for more embedding, i.e. those blocks with higher variance, and vice versa.

In Section 4, the mixture based scheme where there is a random allocation of the QF per block is called a "Mixture-random" method; the scheme with the local variance based QF allocation is called a "Mixture-variance" method. We now discuss another approach for improving the embedding rate, which utilizes the fact that the JPEG compression "attack" is known to the encoder.

3.2 Attack-aware iterative embedding

The biggest factor contributing to the reduction in the embedding rate of YASS is the JPEG compression that the image must undergo before it is advertised. There are errors caused in the embedded bitstream because the 8×8 blocks employed in data hiding does not coincide with the JPEG blocks. Hence the JPEG compression is thought of as an *attack* to the data hiding system, and an erasures and errors correction coding framework[16] is employed to deal with these errors.

The good news in this scenario is that the attack is carried out at the encoder itself before releasing the image, and hence is *known* to the encoder once the hidden image is known. However, the bad news is that the JPEG compression attack is highly correlated with the host signal and hence there is no simple framework to predict the attack (nor its statistics) and account for it beforehand. Hence we explore an iterative embedding procedure, in which we repeat the embedding and attack in an iterative manner with the hope that the system will converge towards lower error rate. Thus we can reduce the raw bit error rate (i.e., error rate without coding), which reduces the required redundancy for the error correcting code thus improving the overall embedding rate.

The embedding algorithm is run in the same way as YASS, followed by the initial JPEG compression, which, as stated earlier, acts as an attack. Then the same hiding procedure is repeated on the resulting attacked images so as to *correct* the errors and erasures caused due to the attack. The raw error rate gets reduced after one such

pass, which then translates to higher information bit embedding rate. Practically, it is observed that there is no significant reduction in raw error rate after one iteration of embedding. The method is referred to as **M1** in Section4.

4. EXPERIMENTS AND RESULTS

The aim of these experiments is to compare the embedding rate of the newly proposed hiding methods (mentioned in Section 3) with the DCT-based YASS system[12] as well as with other competing steganographic algorithms such as F5.[13] We vary the hiding parameters of the schemes such that they yield approximately the same (low) detection accuracy while testing on the same dataset with the same steganalysis techniques.

The steganographic security of our scheme is evaluated against the following blind steganalysis schemes. Note that of the five schemes in,[12] only two are chosen here, because the DCT-based YASS scheme was undetectable even at higher embedding rates, for the other three schemes.

1. **PF-274**: Pevny and Fridrich's 274-dimensional feature vector that merges Markov and DCT features.[6]
2. **Chen-324**: 324-dimensional feature vector, proposed by Chen et al,[9] which is an improvement upon the 39 dimensional feature vector,[8] based upon statistical moments of wavelet characteristic functions.

We conduct the steganalysis experiments on a JPEG image dataset having 4500 images, from the MM270K database*. The images stored in this database have been JPEG-compressed at a quality factor of 75. Half of the images are used for training and the other half for testing. The training and testing sets have the same number of cover and stego images. We train a support vector machine (SVM) on a set of known stego and cover images and use the classifier thus obtained, to distinguish between cover and stego images in the test dataset. The size of the images was generally less than or equal to 512 pixels per dimension; for larger sized images, where an individual dimension exceeded 512 pixels, the images were resized such that the largest dimension equaled 512, while maintaining the image aspect ratio. The results on larger sized images are presented in Section 5.

The steganalysis performance is quantified through the detection accuracy - its computation is discussed in[12] and is repeated here for convenience. 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_d as our evaluation criteria, which is defined as follows.

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

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_d 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_d being close to 0.5 implies nearly undetectable hiding, and as the detectability improves, P_d should increase towards 1.

4.1 Choosing the parameters for evaluation

In order to be able to evaluate the schemes and be able to compare the performance, we set the internal parameters of the various methods such that they yield the same detection rate. The process used is briefly outlined below for the original YASS scheme. Similar approach is used for other schemes that are compared in this paper. When the detection accuracy using a certain steganalysis method is less than 0.60, we consider

*The database was downloaded from http://www-2.cs.cmu.edu/yke/retrieval

the hiding scheme to be statistically undetectable. For computing the embedding rate, we report the number of information bits embedded per non-zero coefficients (bpnc) averaged over 500 images.

As seen in,[12] there are two design parameters under the control of the data hider: the design quality factor QF_h and the output quality factor QF_a, at which the image is advertised. Of these two, we assume that QF_a is to remain fixed at 75 since this is perhaps the most popular quality factor used for JPEG images and hence would not incite minimal suspicion. Thus we can now vary QF_h. As reported in,[12] when QF_h is equal to (or very close to) QF_a, the detection rate P_d is very close to 0.5 (i.e., hiding is undetectable). However the embedding rate is quite low since the JPEG compression introduces a number of errors.

We systematically study how the embedding rate and the detection rate P_d varies with QF_h when QF_a and the big block size B are fixed. The results are reported in Table 1. Note that these results are for the original YASS scheme. It can be seen that the embedding rate is maximum for $QF_h = 50$. However the corresponding detection rate is not low. From this table it is seen that we must set QF_h close to 70 for achieving low detection rates. Hence, we further study the variation in the detection rate with respect to QF_h in the range 65 to 70. These results are shown in Table 2. From this table, we see that in order to ensure that the detection accuracy $P_d \leq 0.60$, we must set $QF_h \geq 69$.

A similar process was used to choose the design parameters for different variants of YASS. Although we discussed the process in detail for the YASS scheme, we will leave out similar description for other methods for brevity.

Table 1. Variation in the embedding rate (in *bpnc*) and detection rate P_d with respect to QF_h. Other parameters are fixed (QF_a is set at 75 and the big block size B is set to 9). It can be seen that the *bpnc* increases as we decrease QF_h, starting from 70, until we get to $QF_h = 50$ after which more coefficients get erased leading to a decline in the embedding rate.

QF_h	40	50	55	60	70
bpnc	0.1941	**0.2031**	0.1956	0.1839	0.1073
PF-274	0.85	0.77	0.69	0.66	0.58
Chen-324	0.86	0.75	0.63	0.61	0.55

Table 2. A closer look at the variation of detection accuracy with respect to QF_h in the range 65 to 70. Similar to Table 1, big-block size B=9 and $QF_a = 75$. It can be seen that for ensuring that the detection accuracy $P_d \leq 0.60$, we need QF_h to be ≥ 69.

Steganalysis Method	Detection accuracy: P_d			
	$QF_h = 65$	QF_h=67	$QF_h = 69$	$QF_h = 70$
PF-274	0.63	0.63	0.60	0.58
Chen-324	0.58	0.58	0.56	0.55

4.2 Mixture-based scheme results

We now present the detection results for the mixture-based YASS scheme discussed in Section 3.1. The setup here is similar to one discussed in the previous section. The QF_a is fixed to 75 and the big-block size is fixed to 9. The other hiding parameters in this schemes are the set of design quality factors used in the *mixture* of the parameters. In these experiments, we use one of three quality factors from, say, 50, 60 or 70. If the choice is made in a random fashion, the scheme is referred to as *50-60-70-rand* to reflect the set of parameters and the selection process. Similarly if the selection is based on the local variance, the scheme is referred to as *50-60-70-var*. The results are reported in Table 3, which compares original YASS with mixture-random and adaptive (mixture-variance) YASS schemes. Table 4 shows similar comparison for a different set of internal parameters. It can be seen that the mixture-variance scheme outperforms the other two.

We now show the effect of the choice of embedding parameters for the mixture-variance schemes. As stated in Section 3.1.1, there are two choices to be made: (i) the set of QF_h to be used (already discussed above), and (ii) the partitions for the variance to determine the particular QF_h to be employed. In Table 5, we study the effect of the choice of different partitions. Note that though the average quality factor reported in this table is

Table 3. Comparison of detection accuracy and embedding rates of the original YASS and the mixture-based schemes. The YASS scheme has the QF_h fixed at 60, while for the mixture based schemes it is ensured that the average quality factor is close to 60. Note that B is set to 9 and $QF_a = 75$. It can be seen that the mixture-variance scheme outperforms the other two.

	YASS (QF_h=60)	Mixture-random (50-60-70-rand)	Mixture-variance (50-60-70-var)
bpnc	0.1839	0.1704	0.1930
PF-274 (P_d)	0.66	0.59	0.59
Chen-324 (P_d)	0.61	0.58	0.58

Table 4. Another comparison of detection accuracy and embedding rates of the original YASS and the mixture-based schemes. In this case, the YASS scheme has the QF_h fixed at 70, while for the mixture based schemes it is ensured that the average quality factor is close to 70. Here also, B is set to 9 and $QF_a = 75$ as in Table 3. It can be seen that the mixture-variance scheme outperforms the other two.

	YASS (QF_h=70)	Mixture-random (60-70-80-rand)	Mixture-variance (60-70-80-var)
bpnc	0.1073	0.0763	0.1365
PF-274 (P_d)	0.58	0.56	0.56
Chen-324 (P_d)	0.55	0.54	0.54

not much meaningful, we compute it and report it here to ensure that the adaptive scheme is not biased towards any one partition.

For the Mixture-variance scheme, the question arises as to what partitioning of the variance range is better for the bpnc-detection trade-off. In Table 5, it is seen that as the size of the last partition is increased, the effective or average QF_h decreases slightly and P_d increases slightly. Thus, there is only a slight trend in the variation of the embedding rate or detection rate with change in the partitions. As the changes are not significant enough, we use a partitioning of $[0, 1, 4, \infty]$ for all the experiments with Mixture-variance in the sections below.

Table 5. The effect of varying the partitioning of the variance values is studied. Here, Avg. QF_h refers to the average QF obtained, after considering the QF allocation over all the blocks. Note that in this table, the interpretation of, say, $[0, 1, 4, \infty]$ where the mixture has QF_h=50,60 and 70, is that a QF_h of 70/60/50 is used for the zone [0,1)/[1,2)/[2, ∞) of block-based variance values, respectively.

partitions	50-60-70	Avg. QF_h	PF-274 (P_d)	Chen-324 (P_d)	60-70-80	Avg. QF_h	PF-274 (P_d)	Chen-324 (P_d)
$[0, 1, 2, \infty)$	0.1952	58.85	0.60	0.59	0.1429	68.02	0.55	0.55
$[0, 1, 4, \infty)$	0.1930	59.61	0.59	0.58	0.1353	68.82	0.54	0.54
$[0, 1, 6, \infty)$	0.1918	60.10	0.59	0.58	0.1323	69.35	0.54	0.54

4.3 Results for YASS-M1 iterative embedding

We now present the results for the iterative scheme that attempts to correct the errors in the hidden bitstream via an additional embedding iteration. The results are reported in Table 6, which also compares the iterative embedding method with all other methods discussed so far (original YASS, mixture-random, as well as the mixture-variance schemes). It can be seen that the iterative embedding gives the best embedding rate among these. The results are provided for two big-block sizes B=9 and B=25. As explained in,[12] the embedding rate increases with B=25, as compared to B=9, at the cost of slight increase in P_d.

4.4 Comparison with F5

We now study the performance of the F5 scheme[13] with the goal of comparing F5 with our schemes. For our dataset, the performance of F5 in terms of embedding rate versus detection rate is given in in Table 7.

Table 6. Comparison of YASS-M1 iterative embedding scheme with other methods - the interpretation of the schemes in the left-most column is as follows: (i) YASS: QF_h=60 refers to the original YASS hiding using a constant QF_h=60, (ii) *50-60-70-rand* refers to hiding using a mixture of QF values, 50, 60 and 70, when the allocation is done randomly, (iii) *50-60-70-rand-M1* refers to the M1 iterative embedding scheme (1 iteration) being used after hiding using *50-60-70-rand* method, (iv) *50-60-70-var* refers to hiding using a mixture of QF values, 50, 60 and 70, when the allocation is done based on local variance, and (v) *50-60-70-var-M1* refers to the M1 iterative embedding scheme (1 iteration) being used after hiding using *50-60-70-var* method.

Hiding Method	B=9 (bpnc)	B=9: PF-274 (P_d)	B=25 (bpnc)	B=25: PF-274 (P_d)
YASS: QF_h=60	0.1839	0.66	0.2073	0.67
50-60-70-rand	0.1704	0.59	0.1907	0.60
50-60-70-rand-M1	0.2335	0.63	0.2552	0.64
50-60-70-var	0.1930	0.59	0.2191	0.60
50-60-70-var-M1	0.2375	0.64	0.2651	0.65

Table 7. Variation of the average detection accuracy with the hiding rate (in bpnc) for F5.

bpnc	0.20	0.18	0.15	0.10	0.08	0.06	0.04
PF-274	0.89	0.86	0.83	0.75	0.67	0.63	0.56
Chen-324	0.64	0.60	0.56	0.53	0.52	0.51	0.50

To enable side-by-side comparison, we reproduce the information from previous tables in Table 8. It can be seen that a higher bpnc than that of F5 can be obtained through the YASS based methods, while the detection accuracy is also reduced. This information is also illustrated through Fig. 1.

Table 8. Comparing the schemes presented in the paper with F5: it can be seen that lower detection rates can be achieved using our schemes for equal or higher embedding rates.

Hiding Method	bpnc	PF-274 (P_d)	Chen-324 (P_d)
F5 (bpnc=0.08)	0.0800	0.67	0.52
F5 (bpnc=0.15)	0.1500	0.83	0.56
B=9, QF_h=60	0.1844	0.66	0.61
B=9, 50-60-70-rand	0.1704	0.59	0.58
B=9, 50-60-70-rand-M1	0.2335	0.63	0.61
B=9, 50-60-70-var	0.1930	0.59	0.58
B=9, 50-60-70-var-M1	0.2375	0.64	0.61

5. DEPENDENCY ON THE IMAGE SIZE AND INITIAL COMPRESSION

The size of the images used for hiding is an important aspect influencing the performance of a steganographic or conversely a steganalytic scheme. This aspect has not been investigated from a practical point of view in the literature. There have been a few theoretical studies exploring the relationship between dimension of the data and the steganalytic detection rate (Moulin and Wang,[17] and Ker[18]). Wang and Moulin[17] use Kobayashi-Thomas bound in context of optimal statistical steganalysis to infer that the detector error for i.i.d. data is $> exp(-N)$, where N is the dimensionality of the covertext or stegotext. Ker[18] shows that for covers of uniform capacity and a quantitative steganalysis method satisfying certain assumptions, the secure steganographic capacity is proportional to the *square root* of the number of covers. In this section we present some preliminary results on our experiments with varying image size.

The results reported in Section 4 have been on images that are JPEG-compressed at a quality factor of 75. Here, we study the detection performance of the YASS steganographic schemes when the image sizes and the input QF (denoted QF_i) are changed. Note that by *input* QF, we mean the QF of the images in the database (used both for hiding and as plain cover). Note that QF_h and QF_a may be different from QF_i.

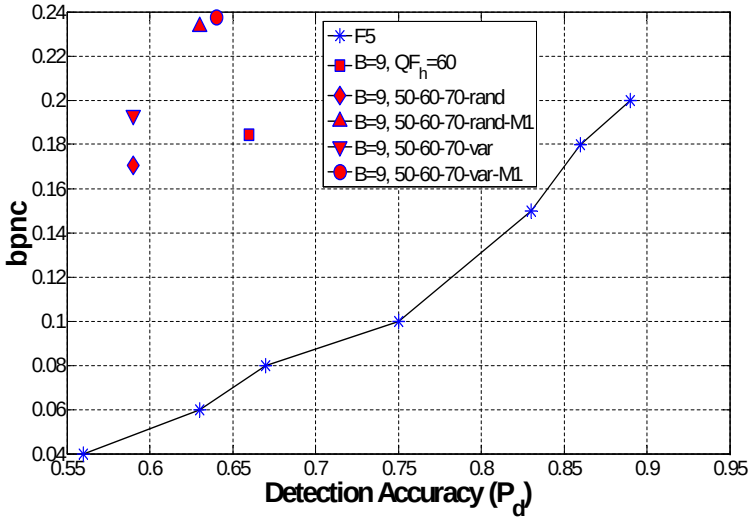

Figure 1. Comparison of bpnc vs P_d performance for different methods: the variants of the YASS scheme clearly outperform the F5 scheme in the bpnc versus detection accuracy trade-off.

In Table 9, we provide the steganalysis results using the PF-274 features for a JPEG image dataset with $QF_i = 95$. The size of the original images are 1600×1200 for some and 2592×1944 for the others. For creating the smaller-sized images (e.g. 512×512 images), we crop out the relevant sized part from the central part of the image. It can be seen that the detection accuracy increases significantly when compared with the QF-75 images used in Section 4, and is near-perfect for the full-sized images. The output quality factor QF_a is set to 75 in this case, similar to all prior experiments. Comparing the results with those presented in[12] as well as those presented in Section 4, it can be seen that QF-95 images are more detectable with respect to QF-75 images, when QF_a is fixed to 75.

Table 9. Hiding performed on QF-95 images of varying sizes (training and testing experiments are done on images of similar size): the average detection accuracy for the different hiding parameters are reported for PF-274 based steganalysis. The full-sized images are of size 1600×1200 or 2592×1944.

Image size	B=12, QF_h=75	B=15,QF_h=75
512 × 512	0.69	0.67
1024 × 1024	0.75	0.71
full-size	0.99	0.99

We conjecture that the poor performance (high detection accuracy) for full-size images in Table 9 is due to the fact that images had originally been compressed at 95, and hence the cover images have double compression artifacts, while the stego images do not. To verify this, we JPEG-compressed all the QF-95 images using QF of 75 before embedding the data. The new results are reported in Table 10. It can be seen that the detection accuracy does decrease substantially, especially for the full-sized images, with B=15, QF_h=75. Thus the data hider would be better placed by compressing the image at the output quality factor before embedding data (i.e., by choosing $QF_i = QF_a$).

6. CONCLUSIONS

The YASS scheme had been previously shown to resist popular blind steganalysis schemes, though the effective embedding rate was low. In this paper, further variants of YASS have been proposed that significantly increase the embedding rate. For the same detection accuracy, YASS has been shown to have a higher embedding rate than the original YASS scheme as well as the matrix embedding based F5 approach. In particular, the iterative embedding (YASS-M1) approach has shown noticeable improvement.

Table 10. Hiding after precompressing the dataset: The same hiding experiments as in Table 9 are repeated (for the same QF-95 images) after compressing them using a QF of 75 before embedding the data. The output quality factor used is QF_a=75. Comparing with the results in Table 9, it can be seen that precompression does reduce the detection rate, particularly for the full-size images.

Image size	B=12, QF_h=75	B=15,QF_h=75
512×512	0.63	0.58
1024×1024	0.72	0.62
full-size	0.91	0.74

An important advantage of the proposed methods as compared to other steganographic schemes (such as F5,[13] perturbed quantization,[19] and matrix embedding based approaches[20]) is that they are inherently robust against distortion constrained attacks due to the use of powerful error correcting codes. This can enable active steganography. Although we have not systematically studied the robustness properties of the presented schemes, the coding framework has been shown to be quite effective in resisting distortion constrained attacks.[16] It may be argued that robustness against attacks is not a true requirement for steganographic schemes. However, we believe that it is important for steganographic methods to not be fragile against mild attacks since an adversary can simply perturb the images slightly (via recompression or any other mild modifications) to thwart any covert communication. In the future, we will study the robustness properties of the presented methods.

ACKNOWLEDGMENTS

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

REFERENCES

1. P. Sallee, "Model-based steganography," in *IWDW 2003, LNCS 2939*, pp. 154–167, Oct. 2003.
2. N. Provos, "Defending against statistical steganalysis," in *10th USENIX Security Symposium*, (Washington DC, USA), 2001.
3. S. Hetzl and P. Mutzel, "A graph theoretic approach to steganography," in *9th IFIP TC-6 TC-11 International Conference, Communications and Multimedia Security*, **3677**, pp. 119–128, (Salzburg, Austria), 2005.
4. K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Provably secure steganography: Achieving zero K-L divergence using statistical restoration," in *Proc. ICIP*, pp. 125–128, 2006.
5. T. Pevny and J. Fridrich, "Multi-class blind steganalysis for JPEG images," in *Proc. of SPIE*, (San Jose, CA), 2006.
6. T. Pevny and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," in *Proc. of SPIE*, (San Jose, CA), 2007.
7. S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *Lecture notes in computer science: 5th International Workshop on Information Hiding*, **2578**, 2002.
8. G. Xuan, Y. Q. Shi, J. Gao, D. Zou, C. Yang, C. Yang, Z. Zhang, P. Chai, C. Chen, and W. Chen, "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.
9. Y. Q. Shi, C. Chen, and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," in *Lecture notes in computer science: 8th International Workshop on Information Hiding*, 2006.
10. C. Chen, Y. Q. Shi, W. Chen, and G. Xuan, "Statistical moments based universal steganalysis using JPEG-2D array and 2-D characteristic function," in *Proc. ICIP*, pp. 105–108, (Atlanta, GA, USA), Oct. 2006.
11. J. Fridrich, T. Pevny, and J. Kodovsky, "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," in *Proc. ACM*, pp. 3–14, Sept. 2007.
12. K. Solanki, A. Sarkar, and B. S. Manjunath, "YASS: yet another steganographic scheme that resists blind steganalysis," in *9th International Workshop on Information Hiding*, Jun 2007. `http://vision.ece.ucsb.edu/publications`.

13. A. Westfeld, "High capacity despite better steganalysis (F5 - a steganographic algorithm)," in *Lecture notes in computer science: 4th International Workshop on Information Hiding*, **2137**, pp. 289–302, 2001.
14. C. Cachin, "An information theoretic model for steganography," *LNCS: 2nd Int'l Workshop on Info. Hiding* **1525**, pp. 306–318, 1998.
15. K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Statistical restoration for robust and secure steganography," in *Proc. ICIP*, pp. II–1118–21, 2005.
16. K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Robust image-adaptive data hiding based on erasure and error correction," *IEEE Trans. on Image Processing* **13**, pp. 1627 –1639, Dec 2004.
17. Y. Wang and P. Moulin, "Steganalysis of block-DCT image steganography," in *IEEE workshop on Statistical Signal Processing*, (St Louis, MO, USA), Sept. 2003.
18. A. D. Ker, "A capacity result for batch steganography," *Signal Processing Letters* **14**(8), pp. 525–528, 2007.
19. J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "Writing on wet paper," in *Proc. of SPIE: Security, Steganography, and Watermarking of Multimedia Contents VI*, pp. 428–445, (San Jose, CA, USA), Jan. 2005.
20. Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion," in *Lecture notes in computer science: 8th International Workshop on Information Hiding*, pp. 314–327, 2006.