Influence of Embedding Strategies on Security of Steganographic Methods in the JPEG Domain

Jan Kodovský, Jessica Fridrich*

Department of Electrical and Computer Engineering, Binghamton University, State University of New York

ABSTRACT

In this paper, we study how specific design principles and elements of steganographic schemes for the JPEG format influence their security. Our goal is to shed some light on how the choice of the embedding operation and domain, adaptive selection channels, and syndrome coding influence statistical detectability. In the experimental part of this paper, the detectability is evaluated using a state-of-the-art blind steganalyzer and the results are contrasted with several adhoc detectability measures, such as the embedding distortion. We also report the first results of our steganalysis of the recently proposed YASS algorithm and compare its security to other steganographic methods for the JPEG format.

1. INTRODUCTION

Steganography is the act of covert communications, which means that only the sender and receiver are *aware* of the secret communication. To accomplish this, the message is typically hidden within innocent-looking objects known as covers. The objective is to embed a secret message so that its very presence in the stego object cannot be proved. Thus, the main requirement of steganography is *undetectability*, which, loosely defined, means that no algorithm exists that can determine whether an object contains a hidden message. Definition of undetectability was provided by Cachin [2] using an information-theoretic framework. Alternative definitions appeared in [19, 33, 15]. The impact of embedding and detection strategies for repeated communication on detectability of the whole stego channel was studied by Ker [21].

1.1. Design elements

In this paper, we study steganographic methods that embed messages in images that are distributed in the JPEG format. All JPEG steganographic techniques that embed messages in the image data can be broadly divided into three categories, each one of which can incorporate a different set of design elements, which are highlighted in italics in the text.

[**JPEG input**] These methods start with a JPEG image, extract the quantized DCT coefficients, modify them in order to embed the secret message, and then reassemble the stego JPEG file. The coefficients are usually determined using a *selection rule* and then a subset of them is modified using a predefined *embedding operation*. Both the selection rule and the embedding operation may be deterministic, probabilistic, dependent on a secret stego-key, or on the cover content. These methods are often combined with *syndrome coding* methods known as *matrix embedding* [7] or *wet paper codes* [11] to decrease the number of embedding changes and/or their embedding impact. For instance, F5 [32], OutGuess [27], Model based steganography [28], and Steghide [16] belong here.

[**Side information**] The second category of embedding methods use *side-information* at the embedder. The methods either require the input image to be in the raw uncompressed format and then embed the message while compressing the image by minimizing the combined distortion due to quantization and embedding [23] or they manufacture the side information by repeated JPEG compression [10]. Depending on the details of their embedding mechanism, these methods may [23] or may not [10] allow the use of matrix embedding. Some require wet paper codes because the selection rule is not available to the decoder.

The main and most serious disadvantage of techniques from the first two categories is that it is possible to approximately estimate the cover image from the stego image using a process called calibration [6], which enables

* Jessica Fridrich: E-mail: fridrich@binghamton.edu, Telephone: +1 607 777 6177, Fax: +1 607 777 4464

relatively reliable steganalysis [26, 29, 12]. Steganalysis attacks based on calibration were the main motivation for the birth of the third (and most recent) category of embedding methods that we describe as robust embedding in an alternative domain.

[**Alternative domain**] The methods from this category embed the message in a different domain robustly (e.g., in the spatial or wavelet domain) and then compress the image at the very end. On the one hand, the JPEG compression masks to a large extent the impact of embedding and the steganalyst can no longer inspect the direct impact of embedding changes. On the other hand, the compression introduces distortion and thus corrupts the message. Thus, the message needs to be embedded robustly so that the payload can be recovered without errors at the receiver. An example of this design element is YASS [30], which uses a QIM-like mechanism to embed the message in selected bands of DCT coefficients of randomly positioned 8×8 blocks. After embedding, the image is compressesed and the stego image is "advertised" as JPEG. Robustness to JPEG compression is achieved by enlarging the payload using repeat-accummulate error correction codes before embedding to guarantee error-free extraction from the compressed image. Obviously, there are many other options how to embed the message robustly in an alternative domain. The main advantage of such methods is that they produce a JPEG stego image with many statistical characteristics appearing "natural" because the image did arise by compressing a raw image. Randomized embedding changes prevent calibration from estimating the cover image, which makes many steganalyzers unable to detect the stego content despite the fact that the embedding distortion is quite large. The embedding efficiency can be as low as 0.25 payload bits per changed DCT coefficient, which is significantly lower than for the techniques from the first two categories. This interesting approach to steganography goes against the established belief that secure steganographic schemes must have high embedding efficiency.

1.2. Design principles

Some of the most compelling design principles for steganography arise from the definition of steganographic security given by Cachin [2], which says that a steganographic scheme is undetectable if the statistical distribution of covers is the same as the distribution of stego objects. Of course, this principle can be invoked in practice only for some appropriately simplified model of covers because the dimensionality of the space of covers is very large for typical mutlimedia files. The steganographer simply tries to embed secret messages while preserving the statistical model of covers [27, 16, 3, 25, 31, 28].

The second principle is a heuristic one. The steganographer tries to embed so that the embedding changes mimic some natural operation or processing, an example of which is image acquisition. Ideas, such as superposition of device noise (stochastic modulation [9, 4]) are specific realizations of this approach. Perturbed quantization [10] attempts to hide messages by using the fact that the process of rounding coefficients whose values are close to the middle of quantization intervals has a random component due to noise present in images. The embedding principle of YASS also belongs here because the uncompressed image is made "a little more noisy" before compression.

The third design principle tries to minimize the embedding distortion [23, 5]. Depending on how it is defined, this approach allows very general formulations. For example, instead of using distortion, one may define the *impact* of making an embedding change at a specific pixel/coefficient and then embed a given payload while minimizing the overall impact [5, 8]. If the embedding impact is correlated with statistical detectability, this principle essentially realizes the Cachin's criterion but in a more manageable manner.

In this paper, we investigate the influence of various design principles and elements on steganographic security. We start in Section 2 with the necessary background information and continue with investigation of individual design elements in Sections 3–5. Section 6 is devoted to steganalysis of the YASS algorithm. The paper is concluded in Section 7.

2. BACKGROUND

The stego image is always in the JPEG format. We describe a JPEG image using a sequence of n integers $\mathbf{g} = (g_1, \ldots, g_n) \in \mathcal{G}^n$, $\mathcal{G} = \{-1023, \ldots, 1024\}$. The steganographic schemes from the first two categories (see Section 1.1) often work with a finite field representation of $\mathbf{g}$ obtained through some symbol-assignment function $symb : \mathcal{G} \to \mathbb{F}_q$. For example, $symb(g_i) = g_i \bmod 2$ ($symb(g_i) = g_i \bmod 3$) assign a bit (ternary symbol) to each

DCT coefficient. Representing the cover image $\mathbf{g}$ as a vector $\mathbf{x} \in \mathbb{F}_q^n$ enables application of matrix embedding and wet paper codes.

We consider as cover image the image on the output of the steganographic method under the condition that no embedding occurred. For example, if a stego method requires a raw image on its input and then embeds the message while compressing it as 75% JPEG (e.g., MMX), then the cover image is the raw image compressed as 75% JPEG. As another example, if YASS accepts 90% JPEG images on its input and then uses the final quality factor $QF_a = 75$ to create the stego image, the cover image is a double-compressed image with primary and secondary quality factors 90 and 75, respectively. We further note that it is important to use the same JPEG compressors for creating the covers as those used in the stego method to avoid artificially increasing the detectability by using different JPEG compressors (see [12] for more details on this issue).

Finally, everywhere in this paper we measure the length of the embedded message in bits per non-zero AC DCT coefficient (bpac) of the cover. This measure of payload relates to the size of the cover and is independent of the embedding capacity of any given steganographic method. Because we use small payloads, we never encountered the problem of not being able to embed a given payload using any stego scheme.

2.1. Evaluating security

We measure the steganographic security through experiments with the blind steganalyzer [26] that uses 274 extended DCT and Markov features merged together. The classifier was implemented using a soft-margin SVM with Gaussian kernel trained on 3500 images and tested on 2500 images. More details about the classifier and its training and testing are in the original publication [26].

All experiments were done on images derived from a database of 6000 images obtained in their raw form from a multitude of different digital cameras and compressed using JPEG. Experiments in Sections 3–5 were carried for a database of single-compressed 80% quality JPEG images with 3.2 megapixels and $538,634$ nonzero DCT coefficients on average. In Section 6, we used a different database whose details are provided directly in Section 6. We would like to emphasize here that any experimental steganalysis results should always be accompanied with information about the size of the images in the database and their JPEG quality because these factors may influence the results of steganalysis quite substantially.

Statistical detectability will be evaluated using the measure previously used in [30, 12]—the probability, P_E, of misclassification for equal prior probabilities of covers and stego images

$$P_E = \frac{P_{FA} + P_{MD}}{2}, \tag{1}$$

where P_{FA} is the probability of false alarms and P_{MD} is the probability of missed detections. While other measures of performance certainly exist (e.g., probability of detection for 0% and 1% of false alarms [24], false alarms at 50% detection [20], or the area under the ROC curve [6]), the specific choice does not matter much here because we are interested in evaluating performance of steganalyzers when their detection success is low ($P_E \approx 50\%$). In this case, all three measures become highly correlated.

The results obtained through the computer experiments are still tied to a specific steganalyzer, which is a clearly undesirable measure of steganographic detectability. Ideally, we would like to measure the detectability using the KL distance in the space of cover images. This is, however, infeasible due to the large dimensionality of the cover space (a class of 4 megapixel images has dimensionality of 4 million). Another possibility is to represent images using features and calculate the KL distance between the samples of cover and stego images directly using non-parametric estimators [1] or statistics proposed for the two-sample problem [14]. We do not pursue these ideas in this paper and instead postpone investigation of such measures to our future work.

3. INFLUENCE OF EMBEDDING OPERATION

We first test the impact of the type of the embedding operation for methods from the first category [JPEG input]. The stego images were prepared by changing a fixed ratio $0 < \beta < 1$ of randomly-chosen non-zero AC DCT

coefficients[†] modified using three different operations: the F5 embedding operation (decreasing the absolute value of coefficients), –F5 operation (*increasing* the absolute value of the coefficients), and ±1 embedding (changing the value by ±1 with equal probability). Before running the tests, we briefly comment on how one may construct embedding schemes from these three operations.

In the original F5 algorithm, the effects of shrinkage (when a DCT coefficient that is equal to 1 or −1 is changed to 0) is alleviated by reembedding the same bit at the next coefficient (because the receiver only reads the message from non-zero DCT coefficients). A much more efficient solution is to let the receiver read the message from *all* coefficients (including those equal to zero) and apply wet paper codes with improved embedding efficiency [11]. As reported in [12], this further decreases the number of embedding changes and significantly improves the security of F5. We call this modified version of F5 the "no-shrinkage F5" or in short nsF5. The nsF5 algorithm can embed relative payload α (in bpac) with β_α changes, where β_α is obtained by dividing the payload in bits, αN_0, by the embedding efficiency e_α available for the relative payload α,

$$\beta_\alpha = \alpha N_0 / e_\alpha. \tag{2}$$

The embedding efficiency e_α as a function of α is shown in Figure 2 in [11].

The shrinkage is completely eliminated for the –F5 embedding operation, which dramatically simplifies the implementation as there is no need to use wet paper codes. Of course, binary matrix embedding can and should be applied.

The ±1 embedding operation also requires wet paper codes to remove the problems with shrinkage. But now, since changes both ways are allowed, we can apply more powerful ternary syndrome codes [7].

Continuing with the description of our experiments, we always changed a fraction β of randomly chosen non-zero DCT coefficients and for each β trained a classifier to distinguish between the clusters of features in the 274-dimensional feature space. The error probability P_E (1) was used to quantify the performance of the classifier (and thus evaluate the embedding operation). The value of β for all three embedding operations was determined from (2) by the number of changes the nsF5 algorithm would exert when embedding relative payloads $\alpha = 0.05$, 0.10, 0.15 and 0.20 bpac. Thus, in this test, we fix our distortion budget and measure the detectability. We stress that the actual embedding schemes will have a different performance due to the syndrome coding schemes they can employ.

Table 1 shows the probability of error (1) for all three methods. As can be seen, the –F5 embedding operation is by far the most reliably detected and thus constitutes the worse type of embedding changes. The ±1 operation is in between F5 and –F5, while the embedding operation of F5 stands clearly as the winner.

Operation/α	0.05	0.10	0.15	0.20
F5	26.31	11.17	5.01	2.29
±1	10.16	2.09	0.46	0.12
–F5	4.01	0.32	0.12	0.06

Table 1. Detector error probability P_E (1) in percents for four change rates β_α(2), $\alpha = 0.05,\ 0.10,\ 0.15,\ 0.20$ bpac, and three types of embedding changes.

This experimental result initiated the following study aimed at explaining this phenomenon. We denote by c_i the DCT coefficients from the cover image after dividing by quantization steps but *before rounding* them to integers. While the range of c_i depends on the implementation of the DCT transform, here we assume c_i are real numbers. The DCT coefficients after rounding are denoted d_i. After applying a steganographic algorithm to the JPEG file, the quantized DCT coefficients d_i are changed to s_i. The coefficients d_i and s_i are thus integers. We denote by $h(m) = |\{i|d_i = m\}|$ the histogram of quantized AC DCT coefficients. For $p > 0$, the distortion due to rounding is

$$D_{round} = \sum_{i=1}^{n} (c_i - d_i)^p$$

[†]The number of all non-zero AC DCT coefficients is denoted N_0.

and the total distortion due to quantization and embedding is

$$D_{emb} = \sum_{i=1}^{n} (c_i - s_i)^p.$$

Theorem: In the absence of any information about the unquantized DCT coefficients, the F5 embedding operation minimizes D_{emb} for any $p > 0$.

Proof: Viewing c_i as instances of a random variable, let $f(x)$ be its probability distribution function. We further assume that f is increasing on $(-\infty, 0)$ and decreasing on $(0, \infty)$ and thus Lebesgue integrable. In practice, $f(x)$ is well approximated with generalized Gaussian distribution [18]. Let us inspect the embedding distortion for one value of the quantized coefficient $d \neq 0$, where, say $d < 0$ (it means that f is increasing at d). The embedding operation will randomly change a fraction δ of $\beta h(d)$ coefficients away from zero (increase their absolute value) and the fraction $1 - \delta$ towards zero (decrease their absolute value). Let us express the increase of distortion due to embedding with respect to the distortion solely due to rounding $D_{emb} - D_{round}$ (follow Figure 1).

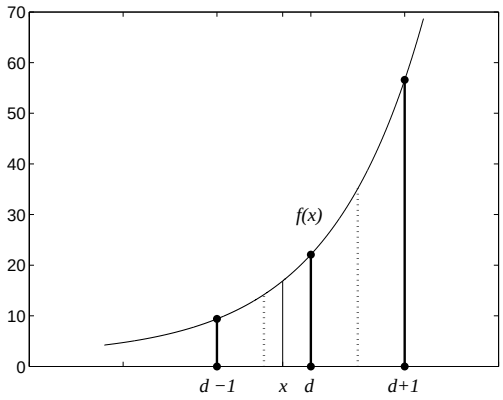

Figure 1. Illustrative example for derivations in the text.

The unquantized DCT coefficients that are quantized to d lie between $d - 1/2$ and $d + 1/2$ (the dotted lines in Figure 1). If an unquantized coefficient $x \in (d - 1/2, d)$ is later changed to $d - 1$ during embedding (with probability δ), the increase in distortion is $d_1(x) = (x - (d - 1))^p - (d - x)^p$. If the coefficient is changed to $d + 1$ (with probability $1 - \delta$), the increase in distortion is $d_2(x) = (d + 1 - x)^p - (d - x)^p$. For coefficients $x \in (d, d + 1/2)$, the increase in distortion is $d_3(x) = (x - (d - 1))^p - (x - d)^p$, when d is changed to $d - 1$ (with probability δ), and $d_4(x) = (d + 1 - x)^p - (x - d)^p$ (with probability $1 - \delta$). Thus, the expected value of the distortion increase, $D(\delta) = D_{emb} - D_{round}$, is

$$\begin{aligned} D(\delta) &= \int_{d-1/2}^{d} (\beta\delta f(x) d_1(x) + \beta(1-\delta) f(x) d_2(x))\, dx + \int_{d}^{d+1/2} (\beta\delta f(x) d_3(x) + \beta(1-\delta) f(x) d_4(x))\, dx \\ &= \delta\beta \int_{d-1/2}^{d} f(x)(d_1(x) - d_2(x))dx + \delta\beta \int_{d}^{d+1/2} f(x)(d_3(x) - d_4(x))dx + C, \end{aligned}$$

where C does not depend on δ. We clearly have $d_1(x) - d_2(x) = d_3(x) - d_4(x) = (x - d + 1)^p - (d + 1 - x)^p = g(x)$. Moreover, $g(d + y) = -g(d - y)$ for $y \in (-1/2, 1/2)$ and $g(x) \geq 0$ on $(d, d + 1/2)$. Thus, we can write for $D(\delta)$

$$D(\delta) = \delta\beta \int_{0}^{1/2} f(d - y) g(d - y) dy + \delta\beta \int_{0}^{1/2} f(d + y) g(d + y) dy + C$$

$$= \quad \delta\beta \int_0^{1/2} \left(f(d+y) - f(d-y)\right) g(d+y) dy + C.$$

Because f is increasing and $g(d+y) \geq 0$ on $(0, 1/2)$, $D(\delta)$ is minimized when $\delta = 0$, which corresponds to the F5 embedding operation. **Q.E.D.**

In the next section, we investigate the influence of the selection channel on the detectability of embedding changes. In particular, we study the influence of texture and position of changes within the 8×8 block.

4. INFLUENCE OF TEXTURE

In this section, we study the impact of several adaptive selection channels on steganographic security. We attempt to answer the following question: Are embedding changes less detectable in textured regions and how much? And if they are, can we construct a good steganographic method by placing the embedding changes in textured areas?

The results reported in [12] indicate that texture does play an important role in Perturbed Quantization steganography [10]. We need to bear in mind, though, that adaptive selection channels usually preclude application of syndrome coding or at least decrease our choices of codes that can be applied. This is because the relative message length increases with decreasing number of *usable* DCT coefficients.

Another very important point is that poor statistical detectability using a blind steganalyzer does not mean that significantly more accurate targeted attacks cannot be constructed. If the selection channel is public, the attacker can focus on areas that were likely modified and use those less likely to have been modified for comparison/calibration purposes. This is a danger of all adaptive techniques that use *public* selection channels.

Having pointed out the caveats of adaptive schemes, we now proceed with the first test in which we study the influence of selection channels determined by local texture in the 8×8 block B. We want the measure to have large values in blocks with complex texture, such as sand, grass, or foliage. In particular, we do not wish to embed into blocks that contain a high contrast edge as such blocks are most likely poor choices for embedding changes. A simple measure compatible with these requirements is calculated in the spatial domain as

$$t(B) = \sum_{(z_i, z_j)} (1 - \delta(z_i, z_j)), \tag{3}$$

where z_i are integer pixel values in block B and δ is an indicator function ($\delta(x, y) = 0$ if $x \neq y$, and $\delta(x, x) = 1$). The sum is over all neighboring pairs (z_i, z_j) within the block. By neighboring, we understand in the horizontal, vertical, and both diagonal directions. In other words, for every pair of neighboring pixels within the block that are different, one is added to the texture measure.

We used the embedding operation of F5 and fixed the number of embedding changes $N_\beta = \beta N_0$ in the same way as in Section 3. The selection channel was determined in the following manner. We calculated the texture measure $t(B)$ for all 8×8 blocks, ordered them from the largest to the smallest, and then made N_β embedding changes in the 10%, 25%, and 50% most textured DCT blocks. The results of embedding into all blocks regardless of texture are also included for comparison. Table 2 summarizes this experiment.

Selection channel/α	0.05	0.10	0.15	0.20
10% most textured blocks	26.32	11.57	5.27	3.23
25% most textured blocks	25.34	10.57	4.69	2.65
50% most textured blocks	25.88	11.71	4.63	2.51
Regardless the texture	26.31	12.21	5.01	2.29

Table 2. Detector error probability P_E (1) in percent for four different change rates in various selection channels determined by block texture (3).

Surprisingly, the statistical detectability does not depend on the block texture much. We need to be careful with the interpretation of the numbers in Table 2, though. The number of changes we made corresponds to what

nsF5 would make using wet paper codes with improved embedding efficiency (2) when embedding payloads α. This assumes that *all* nonzero AC coefficients are allocated for embedding, not only the most textured portion of them. Thus, the detectability of the adaptive schemes will be higher (lower probability of steganalyzer error) because they will have to resort to making more changes for a fixed payload. Consequently, we conclude that texture-adaptive selection channels do not improve steganographic security. It is better to allocate all available DCT coefficients for embedding and instead apply more powerful syndrome coding methods.

This conclusion contradicts our previous findings regarding the influence of texture on PQ methods reported in [12]. We need to keep in mind, though, that in perturbed quantization during JPEG recompression we were able to utilize side information (the single-compressed image). Thus, the embedding changes were always chosen to minimize the combined rounding and embedding distortion. Without this side information, the effect of texture seems to be greatly diminished.

5. INFLUENCE OF SPATIAL FREQUENCY

In this section, we study the influence of the spatial frequency of modified DCT coefficients. Is it better to constrain the embedding changes to a specific frequency band of DCT coefficients? Changes to low-frequency coefficients are multiplied by smaller quantization steps than for high-frequency coefficients. Thus, when it comes to distortion in the spatial domain, constraining ourselves to changes in low-frequency coefficients would incur smaller distortion and lead to smaller detectability using blind steganalyzers. We decided to test this intuitive conclusion.

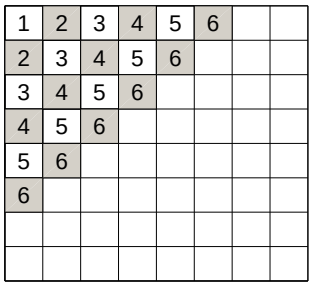

Figure 2. Diagonal subsets within a DCT block.

The 8×8 DCT block was divided into diagonal subsets as shown in Figure 2. Four experiments were performed in which we changed a fixed number N_β of non-zero AC DCT coefficients (again determined by the number of embedding changes nsF5 would do at payloads $\alpha = 0.05,\ 0.10,\ 0.15$, and 0.20 bpac) from the frequency bands 1–3, 2–4, 3–5, and 4–6. We chose three diagonals in order to have enough cofficients for the embedding changes. The results are shown in Table 3. Similar to our study on the influence of texture, we point out that the results are obtained for a fixed distortion budget instead of a fixed message length.

Band/α	0.05	0.10	0.15	0.20
1–3	26.46	13.59	6.50	3.11
2–4	25.96	11.87	5.19	2.53
3–5	20.45	7.40	2.43	0.92
4–6	14.86	5.81	2.83	0.92

Table 3. Detector error probability P_E (1) in percents when making the same number of changes to DCT coefficients from three frequency bands.

Compared with the results of the influence of texture in the previous section, the position within the DCT block where changes are made is more important for statistical detectability. Changing the low-frequency DCT

coefficients is less detectable, which corresponds to our intuition that lower embedding distortion is in fact a good design criterion for methods from the first two categories [JPEG input] and [Side information]. This conclusion contradicts the results reported in [22], where images with higher embedding distortion seemed to have better security. This discrepancy may be due to several reasons. First, we are studying low embedding rates compared to [22] because higher embedding rates can be reliably detected and there is little hope to obtain secure steganography for such rates. Higher embedding rates lead to less reliable calibration [26] because it is harder to estimate the cover image from the more distorted stego image and thus cause the classifier to be less accurate. This is confirmed by the observation made in [22] that classifiers not using calibration did not exhibit this curious phenomenon (higher distortion at lower detectability). Second, the classifier tested in [22] is different (based on 23 DCT features).

The results reported in Table 3 are again obtained for a fixed distortion budget and thus do not correspond to any embedding technique. We decided to simulate the real embedding process this time to see how certain syndrome coding techniques influence detectability. After allocating a band of DCT coefficients, we applied Hamming codes for relative message length N_β/N_{band}, where N_{band} is the number of non-zero AC DCT coefficients in the selected frequency band. The purpose here is to determine what is more important—the more powerful matrix embedding and thus fewer embedding changes or the location of the coefficients we change in the blocks?

Method/N_β	0.05	0.10	0.15	0.20
nsF5 in 1–3	25.18	11.93	6.26	5.25
nsF5 in 1–4	27.77	13.85	7.22	3.25
nsF5 in 1–5	26.68	13.27	6.94	2.93
nsF5 (everywhere)	26.31	11.17	5.01	2.29

Table 4. Error probability (1) in percents when embedding the same payload in different frequency bands using the nsF5 algorithm.

From Table 4, we can see that incorporating the diagonal embedding restriction into the embedding algorithm has a negligible effect on detectability. Moreover, using public selection channels introduces further vulnerabilities to targeted attacks.

6. STEGANALYSIS OF YASS

In this section, we evaluate the security of the recently proposed YASS algorithm [30], which is a representant from the third class of embedding algorithms [alternative domain]. Methods in this category do not embed the message in the DCT domain and therefore the steganalyst can no longer inspect the direct impact of embedding changes. Instead, they embed in a different domain (e.g., the spatial or wavelet domain) and then compress the image at the very end. The JPEG compression masks to a large extent the impact of embedding, but it also introduces errors in the recovered data bits. Thus, the message needs to be encoded using error correction codes to allow the receiver error-free extraction. The coding also provides robustness against active warden who might recompress or filter the stego image.

YASS algorithm utilizes the idea of embedding data in randomized locations in order to disable calibration used in many state-of-the-art steganalytic tools. It uses a QIM-like mechanism to embed the message in selected bands of DCT coefficients of randomly positioned 8×8 blocks which do not coincide with the 8×8 grid used during JPEG compression. After embedding, the image is compressesed and the stego image is "advertised" as JPEG. A different quality factor QF_h can be used for the hiding part of the embedding process than the quality factor QF_a used for the final JPEG compression.

We used the implementation of YASS kindly provided to us by the authors. All experiments were done on 6006 images (see Section 2.1) resized so that their smaller dimension was 512 pixels. The reason for this was to use a database of images with dimensions similar to the images used in [30] in order to be able to compare the results.

In Figure 3, we show the detectability results we obtained under the following experimental setup using the same notation as in [30]: The big block size $B = 9$, the number of low-frequency DCT coefficients used for

embedding in every 8×8 block was 19 and the advertising quality factor $QF_a = 75$. Regarding the type of the input images, we were more restrictive than the authors in [30] to better see the influence of the properties of input images. From the database of resized images, we generated three more databases of JPEG images with quality factors 100, 90 and 75. Together with their original never-compressed raw (PNG) versions, we thus obtained four databases of input images. We applied YASS to all of these datasets with the hiding quality factor $QF_h \in \{65, 70, 75\}$, obtaining thus the appropriate stego images. The covers were generated by processing the input images as in YASS but with skipped embedding. This way, we make sure that we only evaluate the impact of embedding. For example, if YASS accepts 90% JPEG images on its input and then uses the final quality factor $QF_a = 75$, the cover image was a double-compressed image with primary and secondary quality factors 90 and 75. For evaluating the performance, we used the procedure described in Section 2.1, incorporating the steganalyzer with 274 extended DCT and Markov features.

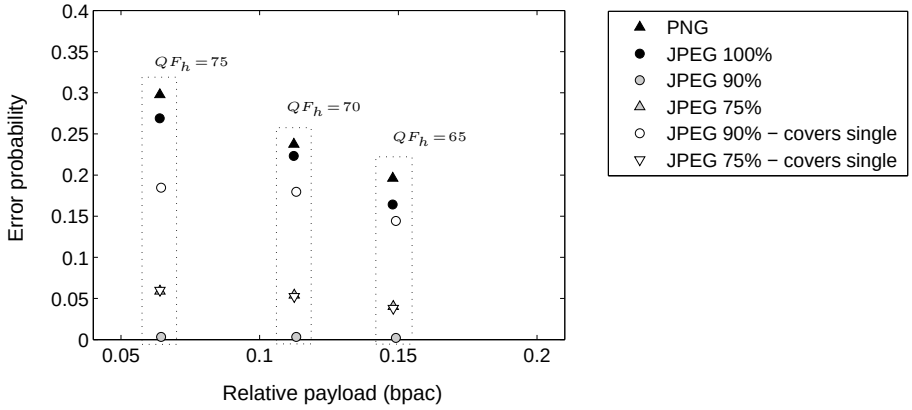

Figure 3. Average detection error (1) of the YASS algorithm with different type of input images and $QF_a = 75$ using 274 extended DCT and Markov features. Also included for the case of 90% and 75% JPEG input images are the detectability results when comparing stego images with single-compressed covers at 75% JPEG quality.

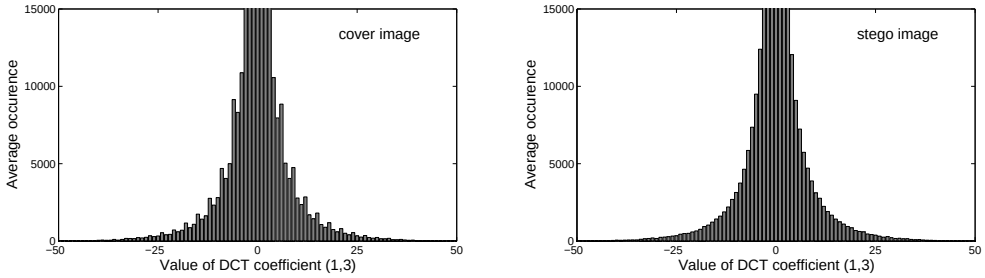

Figure 4. Histogram of DCT coefficients for frequency (1,3) for the double-compressed cover image (left) and YASS stego image (right) for 90% JPEG input images. Both histograms are averages over 100 images.

Figure 3 shows that the detectability substantionally depends on the type of input images. An almost perfect detection was obtained when the input images were 90% quality JPEGs. This is because the cover images exhibit effects of double-compression (Figure 4 left) while the same artifacts are absent in YASS stego images due to the effect of embedding (Figure 4 right). Since stego images do not exhibit traces of double-compression, perhaps it is more appropriate to use as covers for YASS only single-compressed images at $QF_a = 75\%$ quality. After all, this is what a steganalyst would use for testing the image if no traces of double-compression can be found. Thus, we included in Figure 3 the steganalysis results for 90% and 75% JPEG input images this time

with covers as single-compressed 75% JPEG images. Under this modified scenario, the performance of YASS when accepting 90% JPEG images increased rapidly because the steganalysis can no longer rely on the double compression artifacts. On the other hand, the performance of YASS with input images 75% quality JPEG images remained practically unchanged because the stego images in this case do not exhibit traces of double-compression ($QF_h = QF_a$). We note that the numerical values from Figure 3 are shown in Table 5.

At this point, we would like to emphasize that the values of relative payloads (converted to *bpac* with respect to the average number of nonzero AC coefficients in particular image dataset) were announced by authors of the YASS algorithm to us and could not be verified because the implementation of YASS only outputs the length of the encoded bit-stream. The embedding efficiency was calculated as the ratio of payload bits and the number of DCT coefficients in which the cover and stego images differred.

Input	QF_h	P_E	$P_E^\star$	Relative payload (*bpac*)	Embedding efficiency
PNG	65	0.1960	–	0.1479	0.2746
	70	0.2374	–	0.1123	0.2277
	75	0.2977	–	0.0639	0.1429
JPEG 100%	65	0.1641	–	0.1479	0.2746
	70	0.2232	–	0.1122	0.2276
	75	0.2689	–	0.0639	0.1428
JPEG 90%	65	0.0019	0.1441	0.1491	0.2564
	70	0.0031	0.1797	0.1131	0.2128
	75	0.0031	0.1847	0.0644	0.1336
JPEG 75%	65	0.0403	0.0379	0.1479	0.3750
	70	0.0541	0.0523	0.1123	0.3324
	75	0.0589	0.0599	0.0639	0.2271

Table 5. Evaluation of the YASS algorithm with $QF_a = 75$ under different experimental setups. $P_E^\star$ is the SVM detection error when using single-compressed 75% quality JPEG images as covers. More details are provided in the text.

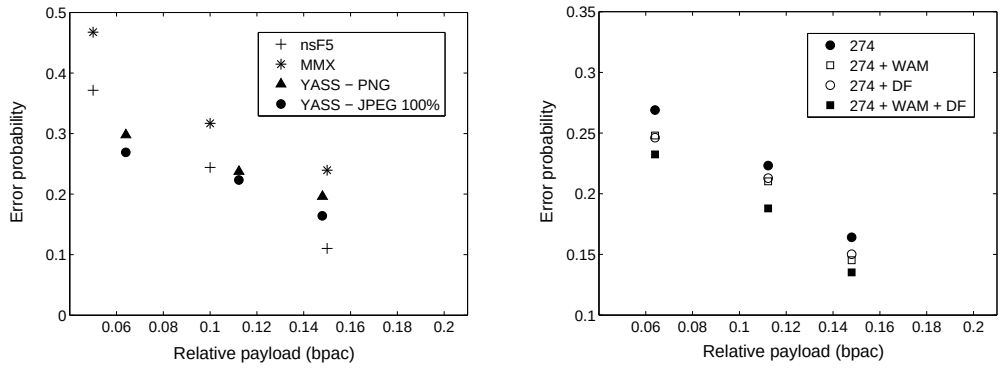

Figure 5. Left: Average detection error (1) of YASS, nsF5, and MMX algorithms using 274 extended DCT and Markov features; Right: Detection error (1) for YASS when incorporating two additional feature sets described in the text.

For the next set of experiments, we restricted ourselves to tests on input images stored as 100% JPEG and PNG for which YASS performed the best. Figure 5 (left) shows the comparison of YASS with selected state-of-the-art steganographic schemes from the first two categories of steganographic schemes. MMX [23] uses side information in the form of the raw version of the input image and incorporates modified matrix embedding by allowing more than one change to minimaze the total embedding distortion. The nsF5 algorithm does not use any side information and accepts JPEGs on its input.

It is certainly quite remarkable that despite the very high number of embedding changes YASS introduces,

its detectability is comparable with the other two methods. This result contradicts the widely believed dogma that secure steganography should minimize the energy of modifications. A disadvantage of YASS is its high implementation complexity when compared with other methods, such as nsF5.

The 274 calibrated extended DCT and Markov features were originally developed specifically for detection of methods from the first two classs [JPEG input] and [Side information]. Because YASS works on an entirely different principle, we decided to expand the feature set to better detect the changes that YASS makes. The first feature set consists of 27 WAM features [17, 13] calculated as higher-order absolute moments of the noise residual in the wavelet domain. These features were used to construct one of the most reliable attacks on ± 1 embedding in the spatial domain and we expect them to be useful for detection of YASS as well because YASS adds noise to the image in the spatial domain. The second feature set we added resulted from an observation of the impact of embedding changes on DCT coefficients. We call this feature set DF (Diagonal Features). It consists of 15 features $DF_1, \ldots, DF_{15}$ calculated as

$$DF_l = \frac{1}{N_0} \sum_{b=1}^{B} \sum_{k=max(1,l-7)}^{min(l,8)} \left[1 - \delta\left(D_b(l+1-k,\ k), 0\right)\right],$$

where l indexes the diagonals ($l = 1, \ldots 15$), B is the total number of 8×8 blocks in the image, $D_b(i,j)$ denotes the quantized DCT coefficient at spatial frequency (i,j), $i = 1, \ldots 8$, $j = 1, \ldots, 8$, in the b-th 8×8 block, N_0 is the number of nonzero DCT coefficients in the image, and δ is the indicator function. In other words DF_l is the number of nonzero DCT coefficients on the l-th diagonal from all 8×8 blocks (normalized by N_0). Figure 5 (right) shows the improvement in detection of YASS (using input images stored as 100% JPEG) after adding the two feature sets to the 274-dimensional feature vector using the same blind steganalyzer as in all previous experiments. The detectability of YASS improved by approximately 5% after adding both feature sets.

7. CONCLUSIONS

In this paper, we study the influence of several design principles and elements commonly employed in steganographic schemes, such as the embedding operation and domain, selection channels determined by block texture and by spatial frequency of the DCT coefficient, and syndrome coding. The operation of decreasing the absolute value of quantized DCT coefficients as employed in F5 provided the best security because in the absence of any side information about the unquantized coefficients, this embedding operation can be shown to minimize the overall distortion due to quantization and embedding.

The influence of texture on detectability of changes is surprisingly almost negligible. The embedding changes constrained to low-frequency DCT coefficients are less detectable, which supports the belief that detectability increases with the energy of embedding modifications.

Our steganalysis of the YASS algorithm [30] indicates that its security highly depends on the type of the input images. The best security (worst detection) was obtained for input images in the never compressed raw format and for 100% quality JPEGs. On these input images, YASS's detectability evaluated using a state-of-the-art blind steganalysis tool was comparable with the performance of no-shrinkage F5 and the MMX algorithm [23]. By introducing two additional feature sets with 27 and 15 features, we were able to further improve the detection of YASS by 5%. Contrasting the high embedding distortion of YASS with its low detectability, it seems that this approach to steganography is quite promising.

References

[1] J. Beirlant, E. Dudewicz, L. Gyorfi, and E. van der Meulen. Non-parametric entropy estimation: An overview. *International Journal of Math. and Stat. Sci.*, 6:17–39, 1997.

[2] C. Cachin. An information-theoretic model for steganography. In D. Aucsmith, editor, *Information Hiding, 2nd International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318. Springer-Verlag, New York, 1998.

[3] J. Eggers, R. Bäuml, and B. Girod. A communications approach to steganography. In E.J. Delp and P.W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, and Watermarking of Multimedia Contents IV, San Jose, CA, January 21–24, 2002*, volume 4675, pages 26–37.

[4] E. Franz and A. Schneidewind. Pre-processing for adding noise steganography. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Proceedings, Information Hiding, 7th International Workshop*, volume 3727 of *Lecture Notes in Computer Science*, pages 189–203. Springer-Verlag, Berlin, 2005.

[5] J. Fridrich. Minimizing the embedding impact in steganography. In J. Dittmann and J. Fridrich, editors, *Proceedings ACM Multimedia and Security Workshop, Geneva, Switzerland, September 26–27, 2006*, pages 2–10. ACM Press, New York.

[6] J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 67–81. Springer-Verlag, New York, 2005.

[7] J. Fridrich, P. Lisoněk, and D. Soukal. On steganographic embedding efficiency. In N. Johnson and J. Camenisch, editors, *Information Hiding, 8th International Workshop*, volume 4437 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 2006.

[8] J. Fridrich and T. Filler. Practical methods for minimizing embedding impact in steganography. In E.J. Delp and P.W. Wong, editors, *Proceedings SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA*, volume 6505, pages 02–03, January 29–February 1 2007.

[9] J. Fridrich and M. Goljan. Secure digital image steganography using stochastic modulation. In E.J. Delp and P.W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents V, Santa Clara, CA, January 21–24*, volume 5020, pages 191–202, 2003.

[10] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography. *ACM Multimedia and Security Journal*, 11(2):98–107, 2005.

[11] J. Fridrich, M. Goljan, and D. Soukal. Wet paper codes with improved embedding efficiency. *IEEE Transactions on Information Security and Forensics*, 1(1):102–110, 2006.

[12] J. Fridrich, J. Kodovský, and T. Pevný. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In *ACM Multimedia & Security Workshop*, pages 3–14, September 20–21 2007.

[13] M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E.J. Delp and P.W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, San Jose, CA, January 16–19*, volume 6072, pages 1–13, January 2006.

[14] Gretton, K. Borgwardt A., M. Rasch, B. Schoelkopf, and A. Smola. A kernel method for the two-sample-problem. Technical report, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2007. MPI Technical Report 157.

[15] J.J. Harmsen and W.A. Pearlman. Capacity of steganalysis channels. In J. Dittmann and J. Fridrich, editors, *Proceedings ACM Meltimedia and Security Workshop, New York, NY, August 1–2, 2005*, pages 11–24.

[16] S. Hetzl and P. Mutzel. A graph–theoretic approach to steganography. In J. Dittmann et al., editor, *Communications and Multimedia Security. 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of *Lecture Notes in Computer Science*, pages 119–128, Salzburg, Austria, September 19–21 2005.

[17] T.S. Holotyak, J. Fridrich, and S. Voloshynovskiy. Blind statistical steganalysis of additive steganography using wavelet higher order statistics. In *Proceedings of the 9th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Salzburg, Austria, September 19–21, 2005.*

[18] A. L. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[19] S. Katzenbeisser and F.A.P. Petitcolas. On defining security in steganographic systems. In E.J. Delp and P.W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IV, San Jose, CA, January 21–24, 2002*, volume 4675, pages 50–56.

[20] A. Ker. A general framework for structural analysis of LSB replacement. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Proceedings, Information Hiding, 7th International Workshop*, volume 3727 of *Lecture Notes in Computer Science*, pages 296–311. Springer-Verlag, Berlin, 2005.

[21] A. D. Ker. Batch steganography and pooled steganalysis. In N. Johnson and J. Camenisch, editors, *Information Hiding, 8th International Workshop*, volume 4437 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 2006.

[22] M. Kharrazi, H.T. Sencar, and N.D. Memon. Cover selection for steganographic embedding. In *Proceedings ICIP, Atlanta, GA*, October 2006.

[23] Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In N. Johnson and J. Camenisch, editors, *Information Hiding, 8th International Workshop*, volume 4437 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 2006.

[24] S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, 2006.

[25] H. Noda, M. Niimi, and E. Kawaguchi. Application of QIM with dead zone for histogram preserving JPEG steganography. In *Proceedings ICIP, Genova, Italy*, September 2005.

[26] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E.J. Delp and P.W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, January 29–February 1*, volume 6505, pages 03–04, January 2007.

[27] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium, Washington, DC*, 2001.

[28] P. Sallee. Model-based steganography. In T. Kalker, I.J. Cox, and Y.M. Ro, editors, *Digital Watermarking, 2nd International Workshop, IWDW 2003, Seoul, Korea, October 20–20, 2003*, volume 2939 of *Lecture Notes in Computer Science*, pages 154–167. Springer-Verlag, New York, 2004.

[29] Y.Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In N. Johnson and J. Camenisch, editors, *Information Hiding, 8th International Workshop*, volume 4437 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 2006.

[30] K. Solanki, A. Sarkar, and B.S. Manjunath. YASS: Yet another steganographic scheme that resists blind steganalysis. In T. Furon et al., editor, *Information Hiding, 9th International Workshop, Saint Malo, France, June 11–13, 2007*, to appear in Lecture Notes in Computer Science. Springer-Verlag, New York.

[31] K. Solanki, K. Sullivan, U. Madhow, B.S. Manjunath, and S. Chandrasekaran. Provably secure steganography: Achieving zero K-L divergence using statistical restoration. In *Proceedings ICIP, Atlanta, GA*, October 2006.

[32] A. Westfeld. High capacity despite better steganalysis (F5—a steganographic algorithm). In I.S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302. Springer-Verlag, New York, 2001.

[33] J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf. Modeling the security of steganographic systems. In D. Aucsmith, editor, *Information Hiding, 2nd International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 344–354. Springer-Verlag, New York, 1998.