Modern Steganalysis Can Detect YASS

Jan Kodovský[a], Tomáš Pevný[b]*, Jessica Fridrich[a]†

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

[b]Agent Technology Center, Department of Cybernetics, Czech Technical University in Prague

ABSTRACT

YASS is a steganographic algorithm for digital images that hides messages robustly in a key-dependent transform domain so that the stego image can be subsequently compressed and distributed as JPEG. Given the fact that state-of-the-art blind steganalysis methods of 2007, when YASS was proposed, were unable to reliably detect YASS, in this paper we steganalyze YASS using several recently proposed general-purpose steganalysis feature sets. The focus is on blind attacks that do not capitalize on any weakness of a specific implementation of the embedding algorithm. We demonstrate experimentally that twelve different settings of YASS can be reliably detected even for small embedding rates and in small images. Since none of the steganalysis feature sets is in any way targeted to the embedding of YASS, future modifications of YASS will likely be detectable by them as well.

1. INTRODUCTION

The steganographic algorithm YASS hides messages in digital images so that the stego images can be distributed in the JPEG format.[15,18] Instead of manipulating quantized DCT coefficients in a JPEG file, YASS starts with a spatial-domain representation of the cover image and embeds the message in this domain robustly using a randomized variation of a high-capacity robust watermarking scheme called Quantization Index Modulation (QIM).[1,17] The embedding proceeds in a block-by-block fashion, where the block positions are controlled using a secret key. This embedding mechanism poses two challenges for steganalysts: the embedding changes are made in a domain that is kept secret and they are further masked by the subsequent recompression. These two design elements combined make it hard for steganalysts to identify statistical quantities that could be used to detect YASS.

The fact that state-of-the-art steganalysis methods of 2007 were unable to detect YASS[18] was rather surprising since YASS deviates from the established paradigm of steganographic embedding where the sender minimizes the embedding impact. Instead, YASS introduces more extensive embedding changes than other JPEG steganography methods.‡ In fact, the embedding efficiency of YASS (payload in bits per unit distortion) can be up to ten times lower in comparison with other embedding algorithms.[9]

The failure of practical steganalysis to detect YASS made researchers rethink their approach to steganography. Perhaps, minimizing the embedding impact was not the right embedding guideline. Or, to counter this opinion, the existence of YASS may simply mean that steganalysts do not have the right features that would capture the statistical impact of YASS. Given the key-dependent character of the embedding changes, the task of finding a better set of features seemed daunting.

We now briefly review previous attempts to steganalyze YASS while pointing out some interesting important findings steganalysts learned from the attacks. A partial success in detecting YASS was reported in[9] even though the detection was not very reliable. A reliable targeted attack on YASS appeared in[11] that capitalized on the fact that the embedding is not sufficiently randomized. The significance of this attack, however, is diminished by the fact that YASS could be easily modified so that such targeted attack would no longer be possible.

---

* Portion of this work was done while Tomáš Pevný stayed at INPG - Gipsa-Lab, 961 rue de la Houille Blanche, 38402, Grenoble, France

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

‡This is essential for YASS because the embedding must be robust to JPEG compression.

A good detection of YASS using general-purpose steganalyzers[7] was obtained using feature sets that *do not* involve calibration,[4] a procedure that was specifically developed for attacking JPEG-domain steganography that manipulates quantized DCT coefficients. Calibration is supposed to make the feature $\mathbf{f}(\mathbf{x})$ extracted from image $\mathbf{x}$ more sensitive to embedding changes while making it less sensitive to the image content. The difference-calibrated feature is

$$\mathbf{f}_{\mathrm{d}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}), \tag{1}$$

where $\mathbf{r}$ is a reference image obtained by decompressing $\mathbf{x}$ to the spatial domain, cropping by a few pixels, and recompressing the cropped image using the same quantization matrix as that of $\mathbf{x}$. The reason why the difference-calibrated 274-dimensional Pevny feature vector[13] cannot detect YASS was explained in,[10] where the authors showed that, in the case of YASS, calibration essentially randomizes the features and significantly hurts the detection. At the same time, using non-calibrated version of the same feature set leads to a much better detection. The same paper proposed a modified process of calibration in which the difference (1) was replaced by the Cartesian product

$$\mathbf{f}_{\mathrm{c}}(\mathbf{x}) = [\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{r})]. \tag{2}$$

This way, the steganalyst leaves it up to the machine learning algorithm to decide whether the reference feature $\mathbf{f}(\mathbf{r})$ is useful or whether it should be ignored. Even though the Cartesian calibration doubles the features' dimensionality, modern statistical learning engines, such as support vector machines, are fairly robust to overtraining and non-informative features. Consequently, Cartesian-calibrated features produce results that are consistently better than the difference-calibrated features (1). A significant advantage of calibrating by Cartesian product is that the calibration seems to *always* help and it helps in detecting YASS as well.

The steganographic algorithm YASS is a welcome contribution to steganography. It made researchers rethink their strategies and it improved steganalysis. In this paper, we argue that modern blind steganalysis tools recently developed for detection of JPEG and spatial-domain steganography are capable of reliably detecting various settings of YASS even for small payloads and small images. We compare the detection rates with other steganography algorithms to put this intriguing steganographic algorithm in perspective. In Section 2, we briefly explain the embedding mechanism of YASS and the process of generating the stego images for our experiments. Then, in Section 3 we describe the steganalysis feature sets used in this paper. The experimental results are reported in Section 4. Finally, the results are interpreted and the paper is concluded in Section 5.

2. YASS OVERVIEW

In this section, we first provide a brief overview of the original YASS algorithm[18] and its modified versions later proposed by its inventors.[15] At the same time, we introduce the notation further used in this paper. Section 2.2 contains the details of twelve YASS settings tested in Section 4. In Section 2.3, we explain how we measured the real payload of YASS.

2.1 YASS overview

The original algorithm[18] can be summarized using the following five steps.

1. The entire message is encoded using Repeat-Accumulate (RA) error correction code.[3]
2. The cover image in its spatial-domain representation is divided into big blocks of $B \times B$ pixels, $B > 8$.
3. In each big block, an 8×8 block is pseudo-randomly selected using a secret key.
4. For every such selected 8×8 block, the embedding follows the algorithm originally described in:[17]
   (a) The block is transformed using a two-dimensional DCT.
   (b) Every real-valued DCT coefficient is divided by the corresponding quantization step from a predefined quantization table (corresponding to the hiding quality factor QF_{h}). No rounding of coefficients is performed at this stage.

(c) A fragment of the encoded message is embedded in a predetermined band of 19 low-frequency AC DCT coefficients using QIM, while skipping all coefficients that quantize to zero by the JPEG quantizer.

(d) The block is decompressed back to the spatial domain and put in its original position within the big block.

5. Finally, the image is compressed using JPEG with the advertising quality factor QF_{a} to obtain the stego image.

The final Step 5 necessitates an aggressive error correction code in Step 1. Because the embedding is not confined to the 8×8 grid of JPEG, steganalysis algorithms that inspect only the quantized DCT coefficients are not very effective against this algorithm.

The extraction algorithm first decompresses the stego JPEG image to the spatial domain, identifies the same 8×8 blocks within the big blocks as during the embedding, extracts the encoded (noisy) message bits, concatenates them, and finally applies the RA error-correcting algorithm to extract the secret message.

In,[15] the authors introduced the following two extensions of YASS whose primary purpose was to increase the embedding capacity:

1. Mixture-based QF_{h} approach. Here, the hiding quality factor QF_{h} varies across the 8×8 blocks either randomly or adaptively based on the block variance or the coefficient count.

2. Attack-aware iterative embedding. The embedding process is repeated several times with the hope to obtain a lower error rate, which, in turn, would increase the embedding capacity.

2.2 Tested YASS settings

In our experiments, we tested twelve different configurations of YASS that included both the original and the extended versions described in Section 2.1. The settings are summarized in Table 1.

Setting	QF_{h}	DBs	B	*rep*	bpac
YASS 1	65,70,75	3,7	9	0	0.110
YASS 2	75	-	9	0	0.051
YASS 3	75	-	9	1	0.187
YASS 4	65,70,75	2,5	9	0	0.118
YASS 5	50,55,60,65,70	3,7,12,17	9	0	0.159
YASS 6	75	-	10	0	0.031
YASS 7	65,70,75	3,7	10	0	0.078
YASS 8	75	-	10	1	0.138
YASS 9	65 70 75	3,7	9	2	0.237
YASS 10	75	-	10	2	0.159
YASS 11	75	-	11	1	0.114
YASS 12	65 70 75	3,7	11	0	0.077

Table 1. Twelve settings for YASS as tested in the paper. The explanation of table columns is in the text.

The column 'QF_{h}' contains the hiding quality factor(s), while 'B' stands for the big block size. Settings 1, 4, 5, 7, 9, and 12 incorporate a mixture of hiding quality factors QF_{h} based on block variance[§]. The decision boundaries (DBs) are shown in the column 'DBs' as reported in.[15] (It appears that the actual choice of DBs does not influence the results much.) Settings 3, 8, 9, 10, and 11 use the attack-aware iterative embedding (the column '*rep*' is the number of iterations). Settings 2 and 6 do not use any extensions and correspond to the

[§]Follow Table 1 for YASS 1: if the block variance is in the interval [0,3), we use $QF_{\mathrm{h}} = 65$ for hiding. If the block variance is in the interval [3, 7), we use $QF_{\mathrm{h}} = 70$. Finally, if the block variance is ≥ 7, $QF_{\mathrm{h}} = 75$ is used.

original version of YASS. The last column, 'bpac', is the average payload in bits per non-zero AC DCT coefficient computed across all images in our database. The method for determining the payload is explained in the next section.

In all our experiments, the input cover images were always in the raw (uncompressed) format and the advertising quality factor was fixed to $QF_{\mathrm{a}} = 75$. With these choices, YASS appears to be the least detectable.[9] Also, all settings used the default choice of 19 AC DCT coefficients for embedding.

2.3 Determining the payload

The embedding algorithm of YASS is very different from other steganographic algorithms in several aspects. One of them is that it is not immediately clear from the cover image how many message bits can be embedded. YASS encodes the message using an RA error-correction code with redundancy factor represented with an integer $q > 0$. However, the value of q depends on the number of nonzero coefficients in the 19 lowest-frequency AC bands of selected 8×8 blocks, and on the actual message bits. Consequently, the value of q that guarantees error-free message extraction heavily depends on the image content.

Furthermore, in the implementation provided by the algorithm inventors, YASS is capable of embedding only the full payload. The papers are silent on what strategy should be chosen when a smaller payload needs to be embedded. Instead of making arbitrary choices on how to deal with this issue, and thus potentially negatively affect the security of the algorithm, we simply assume that YASS always embeds the maximum possible payload for each cover image and setting. Consequently, and due to the content-dependent redundancy factor q, the payload that is actually embedded in every image can be very different.

The original implementation of YASS provided by the authors did not output the real message size. Instead, the output was only in terms of the length of the RA-encoded message, which does not tell much about the real payload. In the past, researchers have used several different approaches to deal with this issue:

1. Do not report the payload (for example in[20]) or report the RA-encoded payload size instead of the real payload (i.e.,[11]). This approach, however, does not allow comparing the security of YASS to other steganographic methods.

2. Take the RA-encoded payload size x and some value of the redundancy factor q and calculate the real payload size x/q. The value of q can be either the same for every image (e.g, take $q = 15$) or randomly taken from some prior distribution on q. The latter choice was used in,[7] where the authors assumed a uniform distribution of q on the interval $[10, 40]$. However, the distribution of q is far from uniform and it also depends on the YASS setup even for the same image (see Figure 1).

3. Take the RA-encoded payload size x and consider some lower and upper bound on q (e.g., $q_1 = 10$ and $q_2 = 40$). Then, perform embedding of other stego methods with payloads x/q_1 and x/q_2. This way, we obtain the upper and lower bound for a fair comparison. This approach was used in,[18] where the authors compared the security of YASS with OutGuess and Steghide.

4. For a given YASS setup, use the average bpac values reported by YASS authors for that particular setup (used, e.g., in[9]). Compare with other methods at that payload. This is not the best choice either because the average bpac values depend on the image database.

In this paper, we used the following approach that is free of arbitrary ad hoc choices and, in our opinion, the most fair one.[¶] The authors of YASS kindly provided us with both the embedding and extraction MATLAB routines for YASS. With these tools, the minimum value of q that guarantees decoding without errors can be determined iteratively.[‖] As a result, for a given image and YASS setup, we will always know the real payload x/q. The average payload is then obtained by averaging x/q over all images in the database (the column 'bpac' in Table 1). To be absolutely precise, the bpac value for each image was computed as a ratio between the payload

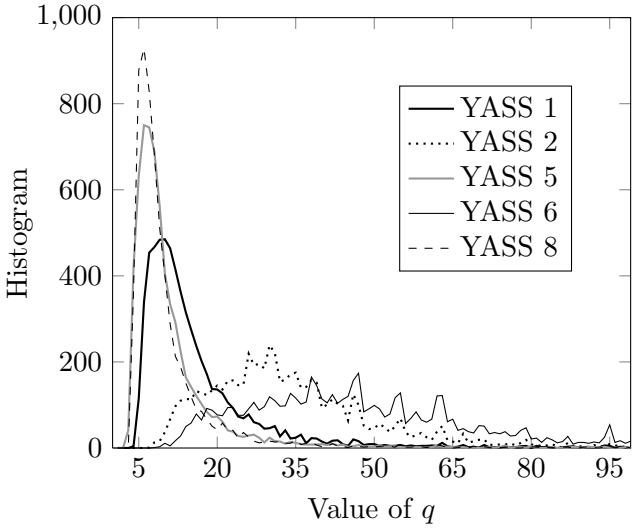

Figure 1. Distribution of q for selected YASS setups. Note that the distributions are highly dependent on the YASS setting and are far from uniform.

x/q and the number of all non-zero AC DCT coefficients in the JPEG file that would result if no embedding took place – the cover image compressed with quality factor QF_{a}.

3. STEGANALYSIS FEATURE SETS

The concept on which YASS is based is quite general and modular in the sense that the specific application may proceed in other domains, such as the wavelet domain, and the robust embedding algorithm may involve other embedding principles than QIM in a subband of block DCT coefficients. The goal of this paper is to steganalyze YASS using blind attacks that do not capitalize on any weakness of a specific implementation of the embedding algorithm. We expect such attacks to be applicable to future variants of YASS that would not suffer from lack of randomization and therefore would resist targeted attacks of the type proposed in.[11]

In the last two years, the reliability of blind steganalysis methods has significantly improved. New feature sets were proposed and existing ones improved. The best detection results in the spatial domain were achieved using feature sets based on modeling differences between neighboring pixels as a first-order Markov model.[21] Additional significant improvement was achieved using higher-order Markov models – the Subtractive Pixel Adjacency Model (SPAM) feature set.[12] The SPAM set is formed by a truncated transition probability matrix of a second-order Markov model of differences between neighboring pixels. This feature set, whose dimensionality is 686, was originally proposed for attacking steganographic algorithms that superimpose on the cover image an independent stego noise in the spatial-domain. From the point of view of how YASS embeds messages, it makes sense to use SPAM to steganalyze it.

The basis of the second feature set used in this paper is the 274-dimensional Pevny feature set,[13] which was proposed for detection of steganographic algorithms that embed messages by slightly modifying the quantized DCT coefficients in a JPEG file. It was originally described in a combination with difference calibration (1). However, according to the recent study,[10] calibrating by difference may significantly hurt the performance for some stego methods, such as YASS. As an alternative, the authors proposed to replace the difference calibration with the Cartesian calibration (see Equation (2)), which always seems to help steganalysis in practice. The

[¶]This approach for evaluating the security of YASS was also used in.[10,15]

[‖]YASS can be used in practice in this manner because the decoder can obtain q from the stego image using the method.[14]

Cartesian calibration doubles the dimensionality of this feature set from 274 to 548. We abbreviate it here by CC-PEV.

The third, Markov Process (MP) feature set is formed by truncated sample transition probability matrices obtained by considering quantized DCT coefficients as a first-order Markov Process (MP) within the DCT blocks and between the blocks.[2] This feature set does not incorporate calibration in any form. The total number of MP features is 486. The MP feature set was the most accurate in detecting YASS as reported in a recent comparative steganalysis of YASS.[7]

In summary, in this paper we steganalyze YASS using the following four feature sets:

- SPAM (686) extracted from the spatial domain. Our expectation is that this feature set may be quite effective against YASS because its embedding is carried out in the spatial domain, but less effective against JPEG-domain steganography.
- CC-PEV (548) extracted mainly from quantized DCT coefficients.
- MP (486) also computed from quantized DCT coefficients. The expectation is that both CC-PEV and MP will likely exhibit similar performance because of their similar pedigree.
- CDF (1,234) (Cross-Domain Feature) set is obtained by merging SPAM and CC-PEV. The goal is to see if there is benefit in merging features computed in different domains.

4. EXPERIMENTS

In this section, we first describe the image database used in all experiments, the machine learning process, and the numerical measure for evaluating the statistical detectability. Then, twelve YASS settings are subjected to steganalysis using four feature sets. Finally, we compare YASS to other steganographic algorithms.

4.1 Image database

The mother image database consists of $6,500$ images acquired in the raw format in their native resolution by 20 different camera models spanning five camera brands. All images from this mother database were converted to 8-bit grayscale and resized using bilinear interpolation so that the smaller side of the image was 512 pixels (aspect ratio preserved). For algorithms that embed in JPEG cover images, all images in the database were compressed with the JPEG quality factor 75, the advertising quality factor of YASS. The average number of non-zero AC DCT coefficients per image was $65,887$.

4.2 Performance evaluation

All steganalyzers were implemented as binary classifiers realized using a soft-margin support vector machine (SVM) with a Gaussian kernel.[16] The image database was randomly divided into approximately two halves for testing and training. The hyper-parameters (C, γ) were optimized using five-fold cross-validation over a fixed grid of values.[6]

The steganalyzer performance is evaluated using the minimal probability of misclassification, $P_{\rm E}$, for equal prior probabilities of cover and stego images

$$P_{\rm E} = \min \frac{P_{\rm FA} + P_{\rm MD}}{2}, \tag{3}$$

where $P_{\rm FA}$ is the probability of false alarms, $P_{\rm MD}$ is the probability of missed detections, and the minimum is taken over the whole ROC curve.

4.3 Results

A separate binary classifier was built for each of the twelve YASS settings and for each steganalysis feature set. The resulting error probabilities P_{E} are reported in Table 2. It is apparent that YASS can be reliably detected by all features. Surprisingly, the detection is far from random guessing even for a very small average payload of 0.031 bpac (see the definition of average payload in Section 2.3).

A closer look at the detection errors in Table 2 reveals that CC-PEV features and MP features exhibit varying (and strongly correlated) reliability when detecting YASS across different setups. On the other hand, steganalyzers based on SPAM features seem to be more stable across different YASS settings. This led us to the idea to combine the DCT-based CC-PEV features with the spatial-domain SPAM features into one large 1,234-dimensional feature set – the Cross-Domain Feature (CDF) set. Table 2 and Figure 2 demonstrate the clear benefit of using the CDF set as it offers the lowest detection error for all twelve settings. This was rather surprising as we anticipated the effect of overtraining to manifest itself given the fact that only 3,250 images are used to train a 1,234-dimensional classifier.

Lastly, due to the very similar performance of MP and CC-PEV feature sets, we did not test the combination MP + SPAM as we expect its performance to be quite similar to that of the CDF.

Algorithm	bpac	MP (486)	CC-PEV (548)	SPAM (686)	CDF (1,234)
YASS 1	0.110	0.110	0.123	0.140	**0.070**
YASS 2	0.051	0.155	0.164	0.152	**0.097**
YASS 3	0.187	0.117	0.086	0.111	**0.055**
YASS 4	0.118	0.098	0.112	0.130	**0.064**
YASS 5	0.159	0.054	0.069	0.094	**0.037**
YASS 6	0.031	0.270	0.260	0.145	**0.124**
YASS 7	0.078	0.237	0.222	0.133	**0.106**
YASS 8	0.138	0.232	0.180	0.121	**0.095**
YASS 9	0.237	0.068	0.046	0.093	**0.028**
YASS 10	0.159	0.202	0.141	0.119	**0.084**
YASS 11	0.114	0.186	0.159	0.178	**0.109**
YASS 12	0.077	0.179	0.194	0.179	**0.135**

Table 2. Steganalysis of YASS using feature sets described in Section 3. The errors P_{E} (3) were averaged over five independent runs with different divisions of the database to training and testing sets. The lowest detection error is in bold.

4.4 Comparative study

In this section, we contrast the statistical detectability of YASS with that of other steganographic algorithms. Due to the lack of time and resources, we focused only on two algorithms – MME3[8] and nsF5.[5] According to the study[5] published in late 2007, MME3 and nsF5 are the least detectable algorithms in the class of algorithms that use side-information at the embedder (the unquantized cover) and algorithms that do not utilize such side-information (their cover is already a JPEG image).

The feature set used for the comparison was the CDF set because it provided the most reliable steganalysis of YASS. Again, a dedicated binary steganalyzer was constructed for each combination of payload and steganographic method. The goal here is to evaluate the statistical distinguishability of cover and stego features for each combination.

The results are displayed graphically in Figure 3 that shows the steganalyzer error probability P_{E} versus the relative payload. Both MME3 and nsF5 are markedly less detectable than YASS and the difference becomes small for payloads larger than 0.15 bpac when all three algorithms become quite detectable ($P_{\mathrm{E}} \lesssim 10\%$). The sudden jumps in P_{E} are due to the effect of matrix embedding used in MME3.

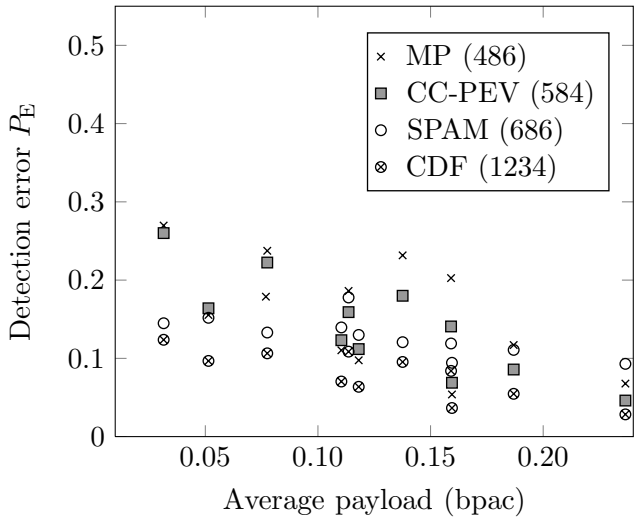

Figure 2. Graphical rendering of Table 2. Steganalysis error P_E for twelve settings of YASS as a function of the average payload in bits per non-zero AC DCT coefficient (bpac).

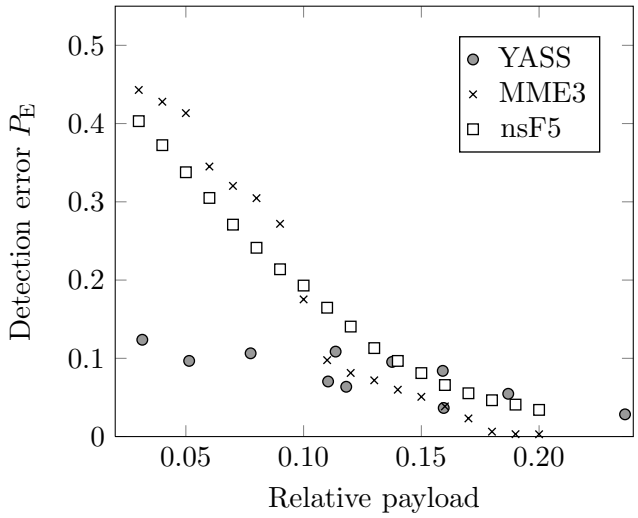

Figure 3. Experimental comparison of the steganographic algorithms YASS, MME3, and nsF5.

One interesting point that surely deserves mentioning is that the difference in statistical detectability between MME3 and nsF5 using the CDF set is rather small. The previously published comparison of these two algorithms using the difference-calibrated PEV features[5] indicated a much bigger difference. While we do not wish to make conclusions from this limited experiment, we wonder if the difference between side-informed steganography and uninformed steganography will disappear with improved steganalysis detectors.

The fact that YASS is more detectable than MME3 and nsF5 validates the established paradigm for designing stegosystems that calls for such embedding mechanisms that minimize the embedding impact.

Finally, we use the opportunity to demonstrate one more important and interesting fact that we observed in our experiments. In Figure 4, we show the detection error versus payload for nsF5 and MME3 for three feature sets. Note that for nsF5 the CDF set exhibits a slightly larger error than CC-PEV. This is most likely due to overtraining caused by the large dimensionality of the CDF set (curse of dimensionality).

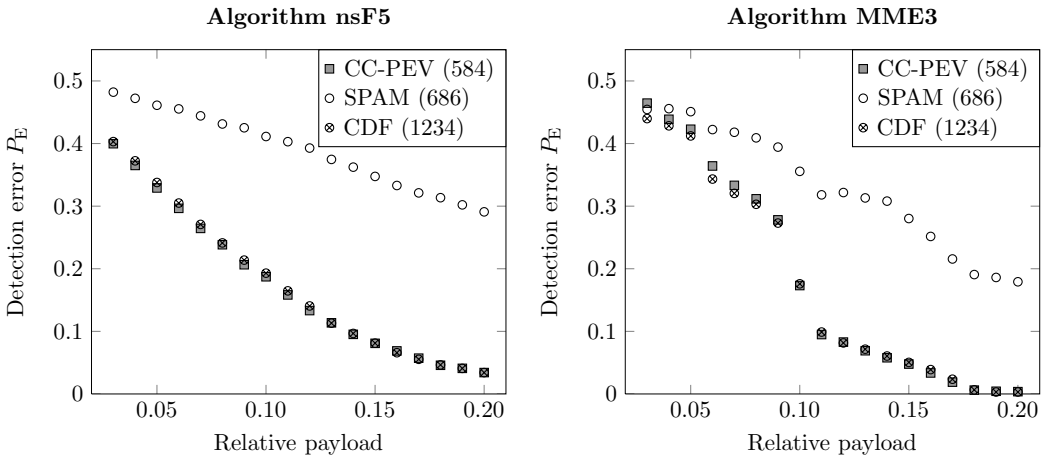

Figure 4. Detection error P_E versus payload for nsF5 and MME3 for CC-PEV, SPAM, and CDF sets.

5. CONCLUSION

The steganographic algorithm YASS is an example of an entirely new philosophy for constructing stego methods. It challenges the long established concept that a necessary condition for a secure method is minimization of the embedding impact. Instead, YASS embeds the message robustly (i.e., with a large distortion) in a key-dependent domain and stores the embedded image as JPEG. The recompression, together with the key-based embedding domain, introduce significant challenges for the steganalyst. The fact that the embedding changes are localized in a domain that is shielded away from the attacker by a key, means that the steganalyst cannot easily identify quantities that are changed by embedding and those that stay invariant. Moreover, the impact of embedding changes is further masked by the recompression. Combined, these two design elements make steganalysis of YASS especially hard. In particular, steganalysis methods that use calibration are rather ineffective in detecting YASS because its embedding mechanism does not directly manipulate the quantized DCT coefficients.

In our opinion, repeatedly stressed throughout this paper, YASS should be viewed as a new embedding paradigm that can be implemented in many different ways, rather than a concrete "trick." Consequently, we intentionally refrained from attacking specific weaknesses of any particular implementation of YASS and, instead, focused on blind attacks using general-purpose feature sets. Such attacks are more likely to be successful against future variants of YASS.

Recently, there has been significant improvement in blind steganalysis of images in the spatial and JPEG domains. Additionally, improved understanding of the process of calibration[10] explained why calibrated feature sets failed to detect YASS and led to an improved calibration procedure, the Cartesian calibration. In the spatial domain, the SPAM feature set,[12] an extension of,[19,21] is a very sensitive and reliable detector of spatial-domain steganography.

In this paper, we used three state-of-the-art steganalysis feature sets – SPAM,[12] the Cartesian-calibrated Pevny feature set,[10] and the Markov process features,[2] and applied them to twelve different settings of YASS. Moreover, we looked at the performance of feature sets that combined features computed in different domains – the 1,234-dimensional Cross-Domain Feature (CDF) set combines the SPAM and the Cartesian-calibrated Pevny set.

In conclusion, the CDF set was able to detect all twelve YASS settings with probability of error $P_E < 15\%$ even for payloads as small as 0.03 bpac and in small images. The fact that the detection was this reliable across various YASS settings is indicative of the fact that other implementations of the YASS paradigm will likely be reliably detected as well.

Moreover, we put YASS in perspective by experimentally comparing its statistical detectability with that of other steganographic methods for the JPEG format – the MME3 and nsF5 methods. These two methods were

chosen as the most secure embedding algorithms in their respective categories (embedding with side-information at the sender and uninformed embedding, respectively). Both MME3 and nsF5 offer significantly better security for payloads up to 0.15 bpac, at which point all three algorithms become quite detectable with $P_{\mathrm{E}} < 10\%$.

Although the goal of this paper was to show that the YASS algorithm can be reliably detected by modern general-purpose steganalyzers, we discovered some other, unrelated findings that we would like to comment upon. The steganalysis of MME3 and nsF5 using the CDF set shows that both algorithms are approximately equally detectable. This is in stark contrast to the steganalysis results published previously using older feature sets.[5] This finding poses an intriguing question of how much the unquantized cover at the sender (the side-information) actually helps in increasing the security as steganalysis continues to improve.

6. ACKNOWLEDGMENTS

The work of Jan Kodovský and Jessica Fridrich on this paper was supported by Air Force Office of Scientific Research under the research grant number FA9550-08-1-0084 and FA9550-09-1-0147. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of AFOSR or the U.S. Government. The work of Tomáš Pevný was supported by the National French projects Nebbiano ANR-06-SETIN-009, ANR-RIAM Estivale, and ANR-ARA TSAR. Finally, we would like to thank Anindya Sarkar for many useful discussion that helped us clarify the embedding algorithm of YASS.

REFERENCES

1. B. Chen and G. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423–1443, 2001.
2. C. Chen and Y.Q. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *IEEE International Symposium on Circuits and Systems, ISCAS*, pages 3029–3032, May 2008.
3. D. Divsalar, H. Jin, and R.J. McEliece. Coding theorems for "turbo-like" codes. In *36th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, September 23–25, 1998.
4. J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 310–323, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.
5. J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.
6. C. Hsu, C. Chang, and C. Lin. *A Practical Guide to Support Vector Classification.* Department of Computer Science and Information Engineering, National Taiwan University, Taiwan.
7. F. Huang, Y.Q. Shi, and J. Huang. A study on security performance of YASS. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2008*, pages 2084–2087, San Diego, CA, October 12–15, 2008.
8. Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 314–327, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.
9. J. Kodovský and J. Fridrich. Influence of embedding strategies on security of steganographic methods in the JPEG domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 2 1–2 13, San Jose, CA, January 27–31, 2008.

10. J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 63–74, Princeton, NJ, September 7–8, 2009.
11. B. Li, Y.Q. Shi, and J. Huang. Steganalysis of YASS. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 139–148, Oxford, UK, September 22–23, 2008.
12. T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 75–84, Princeton, NJ, September 7–8, 2009.
13. 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*, volume 6505, pages 3 1–3 14, San Jose, CA, January 29–February 1, 2007.
14. A. Sarkar, L. Nataraj, B.S. Manjunath, and U. Madhow. Estimation of optimum coding redundancy and frequency domain analysis of attacks for YASS – a randomized block based hiding scheme. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2008*, pages 1292–1295, San Diego, CA, October 12–15, 2008.
15. A. Sarkar, K. Solanki, and B. S. Manjunath. Further study on YASS: Steganography based on randomized embedding to resist blind steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 16–31, San Jose, CA, January 27–31, 2008.
16. B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, 2001.
17. K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Robust image-adaptive data hiding based on erasure and error correction. *IEEE Transactions on Image Processing*, 13(12):1627–1639, Dec 2004.
18. K. Solanki, A. Sarkar, and B. S. Manjunath. YASS: Yet another steganographic scheme that resists blind steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Science, pages 16–31, Saint Malo, France, June 11–13, 2007. Springer-Verlag, New York.
19. K. Sullivan, U. Madhow, S. Chandrasekaran, and B. S. Manjunath. Steganalysis of spread spectrum data hiding exploiting cover memory. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 38–46, San Jose, CA, January 16–20, 2005.
20. X. Y. Yu and N. Babaguchi. Breaking the YASS algorithm via pixel and DCT coefficients analysis. In *Proceedings of the 19th International Conference on Pattern Recognition*, pages 1–4, Tampa, FL, December 8–11, 2008.
21. D. Zo, Y. Q. Shi, W. Su, and G. Xuan. Steganalysis based on Markov model of thresholded prediction-error image. In *Proceedings IEEE, International Conference on Multimedia and Expo*, pages 1365–1368, Toronto, Canada, July 9–12, 2006.