Obtaining Higher Rates for Steganographic Schemes while Maintaining the Same Detectability

Anindya Sarkar[†], Kaushal Solanki[††] and 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
anindya@ece.ucsb.edu,solanki@mayachitra.com,manj@ece.ucsb.edu

Abstract. This paper focuses on modifying the decoder module for an active steganographic scheme to increase the effective data-rate without affecting the embedding module. Three techniques are suggested to improve the error correction framework, an essential component of an active steganographic scheme. The first involves puncturing where the code-length is increased by adding a suitable number of additional erasures. The second technique involves channel modeling and soft-decision decoding which is adaptive to the individual embeddable image coefficient. The third method adjusts the erasure threshold depending on the design hiding quantizer so as to achieve a higher data-rate. Combining these techniques, the effective data-rate is increased by 10%-50% for Yet Another Steganographic Scheme (YASS), a popular active steganographic scheme.

Keywords: data hiding, error correcting codes, puncturing, log-likelihood ratio, erasure rate, steganography

1 Introduction

Steganography is the art of secure communication where the existence of the communication itself cannot be detected by an external agent. The art of detecting such secret communication is known as steganalysis. Covert communication is typically enabled by embedding the secret *message* into an innocuous looking *host* or *cover* signal to form a *composite* or *stego* signal. The task of an adversary, the steganalyst (the "warden"), is to discover the presence of covert communication, which requires use of statistical and/or perceptual analysis to distinguish between plain cover and stego signals. This is the scenario of *passive* steganalysis, wherein, the steganalyst can observe the communication but cannot modify the covers. In many cases, an adversary can simply thwart any covert communication by mildly modifying the signals being communicated without needing to know whether they are cover or stego, leading to what is commonly known as *active* steganography [3, 7, 12]. An active warden has a limited attack budget so as not to significantly affect innocent users, who typically are the majority.

The past decade has seen great strides being made in these competing fields of *steganography* and *steganalysis.* Images are, arguably, the most popular host media, which is evident from the vast amount of literature in image steganography and steganalysis. Blind steganalysis schemes, employing powerful machine learning algorithms and specifically designed image features that capture changes due to data hiding, are quite successful in detecting the presence of very low rate covert communication in image hosts [8, 10, 16].

In this paper we focus on practical aspects of active steganographic schemes, which has received relatively less attention in the literature so far. An active steganographic system can be modeled as a communication channel, wherein, both the data hider and the attacker have limited distortion budgets to modify the host signal. This is in addition to the statistical security that the underlying data hiding scheme must provide. Thus, an important component of a data hiding method that can survive attacks is the use of error correcting codes (ECC). This paper focuses entirely on the error correction aspects of a stego scheme, leveraging ideas from the digital communication literature, so as to provide noticeable improvement in the operating point for the rate-detectability trade-off. We utilize a better modeling of the underlying data hiding channel (scalar quantization index modulation (QIM) [5] based hiding) to compute more accurate likelihood ratios, erasure thresholds, and code rates. In this paper, we consider a benign attack scenario (JPEG compression) for the channel attack.

We employ a popular active steganographic scheme, Yet Another Steganographic Scheme (YASS) [17], as a platform to demonstrate the improvements. YASS involves data embedding in the discrete cosine transform (DCT) domain of randomly chosen block locations. The error correction framework is provided by serial concatenated turbo codes (repeat accumulate codes [6]). The noise channel consists of the JPEG compression attack and our steganalysis framework comprises of a collection of calibrated and uncalibrated features, which was shown to be effective for detecting YASS-based hiding in [9].

Although much of the discussion in the paper is specific to YASS, we must note that if a proper model for the channel attack can be obtained, i.e., if it is possible to reliably estimate a suitable transition probability matrix for the given channel, the methods proposed in the paper can be applied. We do not change the embedder and thus the detectability against steganalysis remains unchanged. *However, by better channel modeling and appropriate modifications to the decoder, the effective data-rate can be increased, without compromising on the undetectability and robustness, which forms the crux of this paper.*

Paper Outline: The problem formulation and main contributions are presented in Sec. 2, followed by a brief overview of the YASS methodology in Sec. 3. A brief description of the composite data hiding channel is presented in Sec. 4. The puncturing scheme and its implications are explained in Sec. 5. Suitable channel modeling for the soft-decision decoding are discussed in Sec. 6. In Sec. 7, we explain how the erasure rate can be suitably varied to maximize the effective data-rate. Experimental results and overall performance improvements over the previous decoding methods are presented in Sec. 8.

2 Problem Setup

QIM-based Hiding Framework: The hiding is performed using quantization index modulation [5]. An embeddable coefficient is converted to the nearest even/odd integer depending on whether the bit to be embedded is 0/1, respectively. For perceptual transparency, we do not use quantized AC DCT coefficients in [-0.5,0.5] for hiding; a DCT coefficient in this range is converted to zero and the corresponding bit is assumed to be "erased". An erasure is denoted as e in the paper. A list of commonly used acronyms is presented in Table 1.

Table 1. Commonly used Acronyms

Acronym	Full form
QIM	Quantization Index Modulation (a commonly used embedding method)
RA code	Repeat Accumulate code (a coding scheme for providing error resilience)
QF	quality factor (determines extent of JPEG compression)
ECC	Error Correction Coding (adds redundancy to data-bits to survive channel attacks)
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
LLR	log-likelihood ratio which denotes the soft confidence level in an embeddable coefficient while decoding
B	big-block size used in the YASS framework
λ	the number of top AC DCT coefficients, encountered during zigzag scan, used for hiding per 8×8 block
λ'	for a puncturing scheme, the hiding band, per 8×8 block, has λ' $(> \lambda)$ coefficients
QF_h	the design quality factor used for hiding
QF_a	the output quality factor at which the stego image is JPEG compressed
q_{opt}	the minimum RA code redundancy factor which allows successful decoding
$[x]$	rounded off value of x
δ_{dec}	at the decoder side, coefficients in $[-\delta_{dec}, \delta_{dec}]$ are assumed to be erasures
N_c	fraction of the embeddable DCT coefficients in the range $[c - 0.5, c + 0.5]$
$N_{[a,b]}$	fraction of the embeddable DCT coefficients in the range $[a, b]$

Turbo codes have shown the most powerful performance for the additive white Gaussian noise (AWGN) and the erasures channels, among many specific channel models. Our approach is practical and the techniques are backed by experimental results and improvements. The following aspects are considered.

(i) **Puncturing:** After encoding the data-bits using a given ECC of known redundancy, the code-length can be further increased by placing erasures at certain locations. Puncturing has long been known to improve the performance of turbo codes [1,2]. It helps the encoding process in two ways: (i) it allows finer choice of embedding rates (rather than $1/q$ where q is a positive integer redundancy factor), and (ii) it allows the use of large codewords, a key factor contributing to the near-capacity performance of turbolike codes. We provide experimental evidence that puncturing increases the effective hiding rate for certain channels .

(ii) **Soft-decision decoding with coefficient-based LLR Allocation:** When the channel model is known a priori, the use of soft-decision decoding invariably improves the convergence probability and accuracy of an ECC decoder. We conduct experimental channel modeling to compute soft confidence values to be set for the log-likelihood ratio (LLR) to be used by the decoder (a sum-product algorithm).

(iii) **Varying erasure rate:** In the QIM-erasure data hiding channel considered in the paper, the decoder can vary a threshold to control the erasure rate. Such a control may not be available in conventional communication channels. We provide simple analysis to compute the optimal value of the threshold δ_{dec} (coefficients in the range $[-\delta_{dec}, \delta_{dec}]$ are assumed to be erased) that maximizes the data hiding rate, given the QIM quantizer (for JPEG case, the quality factor QF_h). These are verified by experiments with real image datasets.

Since the statistical security is demonstrated for the YASS [17] framework, we now provide a brief description of the YASS stego scheme.

3 Brief Overview of YASS

The security of YASS can be attributed to the randomized choice of hiding locations. The idea of YASS was conceived keeping in mind the fact that the steganalysis features (for JPEG images) mainly consist of block-based features, i.e. computed on the 8×8 block. If the hiding is performed using a block-based approach which is not aligned with the JPEG-based blocks, the steganalysis features will be out-of-sync with the features that are modified by the hiding. For hiding in randomly chosen block locations, the image needs to be converted from the compressed domain (if a JPEG image is the input) to the pixel domain.

YASS Framework: The input image is decompressed if it is in JPEG format and then divided into blocks of size $B \times B$ ($B > 8$), where B is called the big-block size. For each big-block, a 8×8 sub-block is pseudo-randomly chosen to hide data. *The encoder and decoder share the same key by which they can access the same set of 8×8 blocks.* For every sub-block, its 2D DCT is computed and then divided by a JPEG quantization matrix at a *design* quality factor (QF), denoted by QF_h. A band of λ AC DCT coefficients lying in the low and mid-frequency range is used for hiding. After data embedding, the resultant image is JPEG compressed at a QF of QF_a.

To emphasize the role of YASS, it decides the hiding locations (pseudo-randomly chosen block locations) and the hiding coefficients (AC DCT coefficients belonging to the hiding band for the randomly chosen blocks). Unless otherwise mentioned, the data-rate computation experiments use B=9, QF_a=75, for the YASS framework.

4 Brief Description of the Hiding Channel

The effective data-hiding channel and the ECC framework used are shown in Fig. 1. We use a repeat accumulate (RA) code [6] as the ECC framework due to the high erasure rate associated with quantized DCT-domain hiding channels.

Composite Hiding Channel: The RA coding framework determines how $\mathbf{u}$, a sequence of N' bits, is mapped to the encoded sequence $\mathbf{c}$, a sequence of $N'q$ bits, assuming the ECC to have a redundancy factor of q. The conversion from $\mathbf{u}$ to $\mathbf{c}$ is explained in Fig. 1. After RA-encoding, this sequence $\mathbf{c}$ acts as the input sequence for QIM-based embedding. For the coefficients lying in the erasure zone, the code-bits get mapped to e (erasures); for the remaining coefficients, the code-bits get properly embedded. The ternary sequence (with symbols $\{0, 1, e\}$) obtained after embedding is denoted by $\mathbf{z}$. Then, the stego image is JPEG compressed (other global noise attacks are also possible) and the same set of embeddable coefficients (as identified at the encoder) is identified at the decoder. The ternary sequence, derived at the decoder side, is denoted by $\hat{c}$. Thus, the effective data-hiding channel is represented by a 2×3 mapping, from $\mathbf{c}$ to $\hat{\mathbf{c}}$.

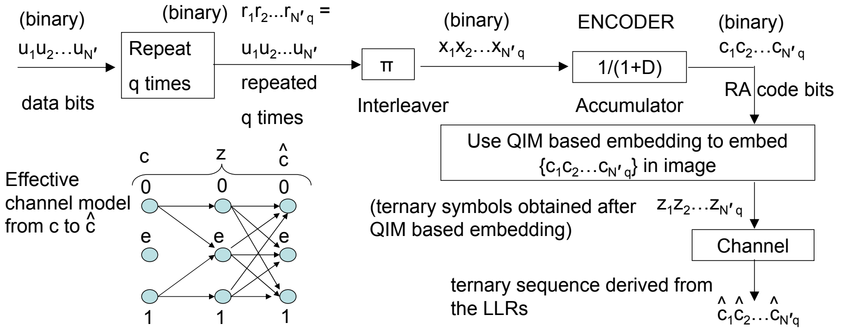
(binary)
$u_1u_2 \ldots u_{N'}$
data bits
(binary)
Repeat
q times
$r_1r_2 \ldots r_{N'q}$ =
$u_1u_2 \ldots u_{N'}$
repeated
q times
π
Interleaver
(binary)
$x_1x_2 \ldots x_{N'q}$
ENCODER
1/(1+D)
Accumulator
(binary)
$c_1c_2 \ldots c_{N'q}$
RA code bits
Use QIM based embedding to embed
$\{c_1c_2 \ldots c_{N'q}\}$ in image
(ternary symbols obtained after QIM based embedding)
$z_1z_2 \ldots z_{N'q}$
Channel
ternary sequence derived from the LLRs
$\hat{c}_1\hat{c}_2 \ldots \hat{c}_{N'q}$
Effective channel model from c to $\hat{c}$
c
z
$\hat{c}$
0
0
0
e
e
e
1
1
1

Fig. 1. The mapping between $\mathbf{c}$, the binary RA code-bit sequence to $\hat{\mathbf{c}}$, the ternary sequence obtained from the LLRs at the decoder output, is shown here for the QIM-RA framework - the "channel" between $\mathbf{z}$ and $\hat{\mathbf{c}}$ refers to the JPEG compression channel that introduces errors and erasures in the mapping from $\mathbf{z}$ to $\hat{\mathbf{c}}$.

Numerical Examples: Some examples of the mapping between $\mathbf{c}$ and $\mathbf{z}$ ($\{0, 1\} \rightarrow \{0, 1, e\}$, denoted by a 2×3 matrix $P_{c,z}$), $\mathbf{z}$ and $\hat{\mathbf{c}}$ ($\{0, 1, e\} \rightarrow \{0, 1, e\}$, denoted by a 3×3 matrix $P_{z,\hat{c}}$), $\mathbf{c}$ and $\hat{\mathbf{c}}$ ($\{0, 1\} \rightarrow \{0, 1, e\}$, denoted by a 2×3 matrix $P_{c,\hat{c}}$), are presented below in Table 2. Since the modifications to the decoding framework require a thorough understanding of the channel transition probability matrices, we present these numerical examples.

Table 2. Using B=9, QF_a=75 for YASS, the transition probability matrices are computed for different hiding conditions. The results are averaged over 250 images.

Hiding Setup	$P_{c,z}$	$P_{z,\hat{c}}$	$P_{c,\hat{c}}$
QF_h=30, λ=6	$\begin{bmatrix} 0.2346 & 0.0000 & 0.7654 \\ 0.0000 & 0.3747 & 0.6253 \end{bmatrix}$	$\begin{bmatrix} 0.9982 & 0.0018 & 0.0000 \\ 0.0011 & 0.9989 & 0.0000 \\ 0.0000 & 0.0002 & 0.9998 \end{bmatrix}$	$\begin{bmatrix} 0.2341 & 0.0007 & 0.7652 \\ 0.0004 & 0.3743 & 0.6253 \end{bmatrix}$
QF_h=75, λ=8	$\begin{bmatrix} 0.4249 & 0.0000 & 0.5751 \\ 0.0000 & 0.6084 & 0.3916 \end{bmatrix}$	$\begin{bmatrix} 0.8333 & 0.1673 & 0.0000 \\ 0.0963 & 0.8880 & 0.0157 \\ 0.0000 & 0.0792 & 0.9208 \end{bmatrix}$	$\begin{bmatrix} 0.3557 & 0.1317 & 0.5126 \\ 0.0602 & 0.5503 & 0.3895 \end{bmatrix}$

The effective 2×3 mapping from $\mathbf{c}$ to $\hat{\mathbf{c}}$ is used to compute the channel capacity $\mathcal{C}$ by maximizing the mutual information $I(\mathbf{c}, \hat{\mathbf{c}})$ between the sequences $\mathbf{c}$ and $\hat{\mathbf{c}}$ (1) - a discrete memoryless channel is assumed.

$$\mathcal{C}_{c,\hat{c}} = \max_{p(c)} \; I(\mathbf{c}, \hat{\mathbf{c}}) = \max_{p(c)} \sum_{c \in \{0,1\}} \sum_{\hat{c} \in \{0,1,e\}} p(c, \hat{c}) \log \left(\frac{p(c|\hat{c})}{p(c)} \right) \tag{1}$$

Given an image, is it possible to obtain these transition probability matrices without actually simulating the entire channel? We now express these matrices in terms of $N_{[a,b]}$, the fraction of embeddable quantized DCT coefficients in $[a, b]$.

Table 3. Expression for a 2×3 transition probability matrix $P_{c,z}$

$p_{0,0} = 1 - p_{0,e}$	$p_{0,1} = 0$	$p_{0,e} = \dfrac{(N_{[-0.5,0.5]} + N_{(0.5,1)} + N_{(-1,-0.5)})}{2}$
$p_{1,0} = 0$	$p_{1,1} = 1 - p_{1,e}$	$p_{1,e} = \dfrac{(N_{[-0.5,0.5]})}{2}$

Explaining $P_{c,z}$: (as shown in Table 2) When a lower QF_h is used, the quantization applied to the DCT coefficients is coarser and hence, the fraction of embeddable DCT coefficients that lies in the erasure zone increases. Therefore, the $(1,3)^{th}$ $(0 \rightarrow e$ mapping) and $(2,3)^{th}$ $(1 \rightarrow e$ mapping) elements in $P_{c,z}$ are higher for QF_h of 30 (coarser JPEG quantization) than when $QF_h = 75$ (finer JPEG quantization), as seen from Table 3.

Why is $P_{c,z}(1,1) < P_{c,z}(2,2)$ (i.e. $p_{0,0} < p_{1,1}$) ? All quantized DCT coefficients in the range [-0.5,0.5] get mapped to erasures. For a coefficient in [-0.5,0.5], it is equally likely for the input bit to be 0 or 1. For DCT coefficients in the range (0.5,1) and (-0.5,-1), the coefficients get mapped to 1 and -1, respectively, when 1 is to be embedded; however, these coefficients are mapped to zero (erasures) when 0 is to embedded. Hence, $p_{1,1} > p_{0,0}$.

We now express the transition probability terms in $P_{z,\hat{c}}$ in terms of the noise distribution $Pr(n)$ (n is the noise signal that affects the mapping from $\mathbf{z}$ to $\hat{\mathbf{c}}$) and the decoder-side erasure cutoff δ.

$$p_{0,0} = \forall_{k \neq 0, k \in \mathbb{Z}} \sum_{x=2k}^{2k+0.5} N_x.Pr(n \leq ([x] + 0.5 - x)) +$$

$$\forall_{k \neq 0, k \in \mathbb{Z}} \sum_{x=2k-0.5}^{2k} N_x.Pr(n \geq ([x] - 0.5 - x)),\ p_{0,1} = 1 - p_{0,0},\ p_{0,e} = 0$$

$$p_{1,1} = \forall_{k \neq 0, k \in \mathbb{Z}} \sum_{x=2k-1}^{2k-0.5} N_x.Pr(n \leq ([x] + 0.5 - x)) +$$

$$\forall_{k \neq 0, k \in \mathbb{Z}} \sum_{x=2k+0.5}^{2k+1} N_x.Pr(n \geq ([x] - 0.5 - x))$$

$$+ \sum_{0.5}^{1} N_x.Pr(n \geq (\delta - x)) + \sum_{-1}^{-0.5} N_x.Pr(n \leq (-\delta - x))$$

$$p_{1,e} = \sum_{0.5}^{1.5} N_x.Pr(n \leq (\delta - x)) + \sum_{-1.5}^{-0.5} N_x.Pr(n \geq (-\delta - x))$$

$$p_{1,0} = 1 - (p_{1,1} + p_{1,e})$$

$$p_{e,e} = \sum_{0}^{0.5} N_x.Pr(n \leq (\delta - x)) + \sum_{-0.5}^{0} N_x.Pr(n \geq (-\delta - x))$$

$$p_{e,1} = 1 - p_{e,e},\ p_{e,0} = 0$$

Explaining $P_{z,\hat{c}}$: (as shown in Table 2) Focussing now on $P_{z,\hat{c}}$, the question comes up as why this matrix is asymmetric? It is observed that $p_{e,1} > p_{e,0}$, and $p_{0,1} > p_{1,0}$. For coefficients in the range [-0.5, 0.5], channel noise can result in the coefficients being rounded off to ±1. Therefore, $p_{e,1} > p_{e,0}$ (the noise value should be significantly high to shift a coefficient from the range [-0.5,0.5] to a range (1.5,2.5) or (-1.5,-2.5)).

For a coefficient to be mapped from $1 \rightarrow 0$, coefficients in the range [0.5,1.5] (or [-1.5,0.5]) can get mapped to (1.5,2.5) (or (-2.5,-1.5)). When a coefficient corresponds to a $0 \rightarrow 1$ mapping, a coefficient in the range [1.5,2.5] (or [-2.5,-1.5]) can get mapped to [0.5,1.5] or [2.5,3.5] (or [-1.5,0.5] or [-3.5,-2.5]). There is very low probability of a '0' getting mapped to an erasure, i.e. to the [-0.5,0.5] zone. On the other hand, it is more likely for a $1 \rightarrow e$ mapping to occur. This happens when a coefficient in [0.5,1.5] (or [-1.5,-0.5]) gets mapped to (-0.5,0.5).

5 Puncturing for Better Performance

Puncturing is a technique used to obtain a $\frac{m}{n}$ code from a "basic" rate $\frac{1}{2}$ code. Puncturing (code bit deletions) effectively decreases the code-length. E.g. when we have a RA codeword of 200 bits and the optimal redundancy factor q_{opt} is 4, we can embed 200/4 = 50 data-bits. The effective codeword length is now increased to 300; the extra 300-200=100 bits are assumed to be erasures. *With respect to puncturing, the input is a 300-bit code-word and 100 bits are deleted from it, leaving a 200-bit code-word.* If the new value of the optimal redundancy factor ≤ 5, the new data rate will be increased as $\lfloor 300/5 \rfloor > 50$. When puncturing is done, the number of additional erasures needs to be decided; it is seen that the effective data-rate is increased for a certain range of additional erasures. Simply put, puncturing allows us to choose a finer embedding rate for a given setup.

The hiding band consists of λ AC DCT coefficients per 8×8 block. Let the number of $B \times B$ blocks used for hiding be N_B. Thus, the total number of coefficients available for hiding $N = \lambda . N_B$. While puncturing, it is assumed that the hiding band consists of λ' ($\lambda' > \lambda$) coefficients per 8×8 block.

We empirically observe that for more noisy channels, performance improvement is not obtained on using a higher number of erasures. The experimentally computed q_{opt} using RA codes is significantly higher than the redundancy factor for an ideal channel code, $\lceil \frac{1}{\mathcal{C}_{c,\hat{c}}} \rceil$ (1). The average value of $(q_{opt} - \lceil \frac{1}{\mathcal{C}_{c,\hat{c}}} \rceil)$, computed for 250 images, is reported in Fig. 2. It is seen that the RA code performs closer to capacity ($(q_{opt} - \lceil \frac{1}{\mathcal{C}_{c,\hat{c}}} \rceil)$ becomes smaller) for larger λ, i.e. for longer code-lengths. It is seen that the RA code is most away from capacity for channels with very high error rates (when QF_h=75) or very high erasure rates (when QF_h=30). In Fig. 2, "I-LLR" and "C-LLR" refer to the "image-dependent" and "coefficient-dependent" LLR allocation methods, explained later in Sec. 6.1 and Sec. 6.2, respectively.

To vary the effective noise level in the channel (this affects the 3×3 transition probability matrix that denotes the mapping between $\mathbf{z}$ and $\hat{\mathbf{c}}$), the design quality factor used for hiding, QF_h, is varied. With a fixed value of QF_a, the output JPEG quality factor, the effective channel noise increases as QF_h is increased. Why does the effective noise increase as QF_h approaches QF_a? As QF_h increases, the DCT coefficients are divided element-wise by a finer quantization matrix (the elements in the JPEG quantization matrix become smaller). For a quantization matrix coefficient of Δ, the noise should exceed $\frac{\Delta}{2}$ to cause a decoding error. Therefore, as $QF_h \Uparrow \implies \Delta \Uparrow \implies$ noise robustness $\Downarrow$.

The bpnc improvements on using varying degrees of additional erasures for different hiding conditions are shown in Tables 4 and 5. For QF_h=70 and 75, the bpnc starts to decrease as $\lambda' > \lambda$, and hence, the corresponding bpnc results after erasure addition are not reported.

6 Suitable LLR Allocation for Soft Decision Decoding

RA codes are an example of serial concatenated turbo codes, where the component decoders are based on the BCJR algorithm [4]. The BCJR algorithm takes as input the a-posteriori probability of each code-bit, which is used to compute the log-likelihood ratio (LLR) at each code-bit location, as defined in (2). The forward and backward Viterbi-decoding algorithms running through

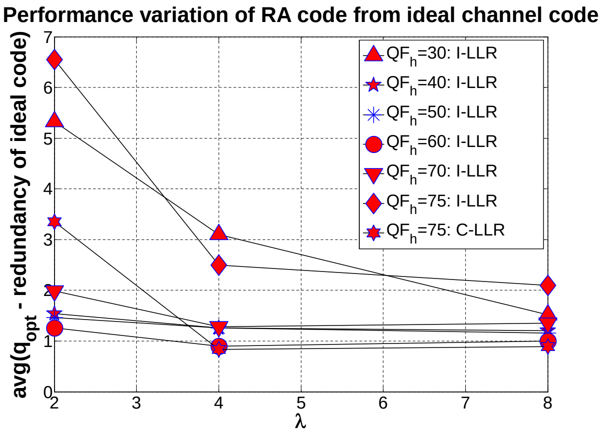

Fig. 2. The variation of the performance with λ and QF_h is computed over 250 images. Here, "I-LLR" and "C-LLR" refer to the "image-dependent" and "coefficient-dependent" LLR allocation methods, respectively.

Table 4. The average bpnc values are computed over 250 images using the QIM-RA framework and puncturing. Here, $(2, 6)$ denotes that λ=2 and λ'=6. The %-gain is expressed as $\dfrac{\max_{\lambda' \geq \lambda}\ \text{bpnc}(\lambda, \lambda') - \text{bpnc}(\lambda, \lambda)}{\text{bpnc}(\lambda, \lambda)}$. We use B=9, δ_{dec}=0.5 and QF_a=75.

$QF_h = 30$						$QF_h = 40$					
(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	%-gain	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	%-gain
0.0318	0.0380	**0.0425**	0.0422	0.0395	**30.50**	0.0526	0.0628	**0.0644**	0.0628	0.0580	**22.43**
(4,4)	(4,5)	(4,6)	(4,7)	(4,8)	%-gain	(4,4)	(4,5)	(4,6)	(4,7)	(4,8)	%-gain
0.0586	0.0687	**0.0732**	0.0705	0.0691	**24.91**	0.0885	0.0969	**0.0999**	0.0989	0.0977	**12.88**
(8,8)	(8,9)	(8,10)	(8,12)	(8,14)	%-gain	(8,8)	(8,9)	(8,10)	(8,12)	(8,14)	%-gain
0.1169	0.1189	**0.1213**	0.1201	0.1170	**3.76**	0.1390	0.1400	0.1416	**0.1449**	0.1417	**4.24**
(12,12)	(12,13)	(12,14)	(12,15)	(12,18)	%-gain	(12,12)	(12,13)	(12,14)	(12,15)	(12,18)	%-gain
0.1387	0.1395	**0.1401**	0.1398	0.1390	**1.01**	0.1644	0.1669	**0.1690**	0.1681	0.1672	**2.80**

Table 5. The same experiments, as in Table 4, are now shown for QF_h of 50 and 60.

$QF_h = 50$						$QF_h = 60$					
(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	%-gain	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	%-gain
0.0508	0.0609	**0.0630**	0.0607	0.0572	**24.02**	0.0427	**0.0446**	0.0421	0.0367	0.0345	**4.45**
(4,4)	(4,5)	(4,6)	(4,7)	(4,8)	%-gain	(4,4)	(4,5)	(4,6)	(4,7)	(4,8)	%-gain
0.0932	0.1016	**0.1040**	0.1032	0.1019	**11.59**	0.0865	0.0908	**0.0937**	0.0917	0.0884	**8.32**
(8,8)	(8,9)	(8,10)	(8,12)	(8,14)	%-gain	(8,8)	(8,9)	(8,10)	(8,12)	(8,14)	%-gain
0.1485	0.1508	**0.1540**	0.1536	0.1481	**3.70**	0.1427	0.1438	**0.1464**	0.1415	0.1373	**2.59**
(12,12)	(12,13)	(12,14)	(12,15)	(12,18)	%-gain	(12,12)	(12,13)	(12,14)	(12,15)	(12,18)	%-gain
0.1748	0.1772	0.1792	**0.1811**	0.1787	**3.60**	0.1738	0.1739	**0.1740**	0.1738	0.1680	**0.12**

the trellis depend on the initial estimates of the posterior probabilities which decide the LLR values. We have consolidated upon a recently proposed method to suitably initialize the LLR estimates at the decoder locations [15]. It has been experimentally observed that proper initialization of LLR values leads to faster convergence at the decoder, i.e. convergence using a lower redundancy factor. For a given image and an attack channel, the LLR values belonged to the 3-tuple, $(\alpha, -\alpha, 0)$, corresponding to 0, 1, and e, respectively, where α is a soft confidence value decided based on the composite channel parameters $(P_{c,\hat{c}})$. We repeat the discussion from [15] in Sec. 6.1 for ease of understanding. This LLR allocation scheme works well, except for very noisy channels. For such channels, we present a per-coefficient based, instead of a per-image based, LLR allocation method in Sec. 6.2.

6.1 Image-based LLR Allocation

Let a certain image coefficient be equal to y and the corresponding embedded bit be b. The LLR value $LLR(y)$ denotes the logarithm of the ratio of the likelihood that a 0 was transmitted through that coefficient $(Pr(b=0|y))$ to the likelihood that a 1 was transmitted $(Pr(b=1|y))$.

$$LLR(y) = \log\left(\frac{Pr(b=0|y)}{Pr(b=1|y)}\right) \tag{2}$$

Let p_e denote the effective error probability in the channel (mapping from $\mathbf{c}$ to $\hat{\mathbf{c}}$) and N_c denotes the fraction of embeddable DCT coefficients whose value changes to c on rounding. In [15], the LLR value is estimated as follows:

$$LLR(y|[y]=c, c\neq 0) = \pm\log\left(\frac{(N_c + N_{c-1}/2 + N_{c+1}/2)(1-p_e)}{(N_c + N_{c-1} + N_{c+1} + N_{c-2}/2 + N_{c+2}/2)p_e/2}\right), \tag{3}$$

where the ± signs are for $c =$ even/odd, respectively, and
$LLR(y)$ is kept at 0 when $[y]=0$.

The distribution of the AC DCT coefficients has been approximated as Laplacian [11, 14]. Always, $N_{c-1} > N_c > N_{c+1}$ holds, for $c \geq 1$, and $N_c \approx N_{-c}$, by symmetry. If we assume $N_c \approx (N_{c-1}+N_{c+1})/2$, then $LLR(y)$ reduces to:

$$LLR(y|[y]=c, c\neq 0) = \pm\log\left(\frac{1}{p_e} - 1\right) \tag{4}$$

It is experimentally observed that the LLR allocation methods using (3) and (4) result in similar embedding rates. Hence, in subsequent experiments, the image-dependent LLR is computed using the relatively simpler expression (4). The next issue is computing p_e for a given image and noise channel. In [15], it is seen that knowledge of the image histogram (that governs $P_{c,z}$) and the output JPEG QF (that governs $P_{z,\hat{c}}$) helps in accurate estimation of p_e.

6.2 Coefficient-based LLR Allocation

This LLR allocation provides the same 3-tuple of LLR values $\left\{\pm\log\left(\frac{1}{p_e}-1\right), 0\right\}$ for all the embeddable coefficients in a certain image. For very noisy channels, we observe that q_{opt} (the minimum RA-code redundancy factor for a given image and a hiding channel) is far-off from the $\lceil\frac{1}{\mathcal{C}_{c,\hat{c}}}\rceil$. The solution is to perform a more in-depth analysis of the LLR allocation. We decide whether a 0/1 is embedded based on the fact that a certain image coefficient rounds off to an even/odd integer. E.g. if a coefficient in the received image is valued at 4, we are "more confident" that it corresponds to a 0-embedding than if the coefficient were valued at 3.6 or 4.4. Denoting the JPEG compression introduced noise signal by n, for the coefficient valued at 4 to correspond to a bit-error, the noise should exceed (4-3.5)=0.5 in magnitude. When the corresponding coefficients equal 3.6 (or 4.4), a bit-error can be caused when the noise exceeds 0.1 in magnitude. Due to the highly Laplacian-like pdf of n, $Pr(n > 0.5)$ is significantly less likely than $Pr(n > 0.1)$. *We have experimentally observed that using a per-coefficient LLR allocation scheme is more advantageous than using a per-image LLR allocation.*

Let y be the received DCT coefficient, $(x-1)$ be an even integer and $[y] = (x-1)$. For a decoding error to occur, the noise signal should exceed $((x-0.5)-y)$; then y would get mapped to x. It is assumed, due to the Laplacian pdf, that the value of the noise signal n is generally limited to [-1,1].

$$LLR(y) = \pm\log\left(\frac{Pr(b=0|y)}{Pr(b=1|y)}\right) = \pm\log\left(\frac{1-Pr(n>(x-0.5-y))}{Pr(n>(x-0.5-y))}\right) \quad (5)$$

where $[y] = (x-1)$, the ± signs are for $(x-1)$ being an even/odd integer, respectively, and $LLR(y)$ is 0 when round$(y) = 0$.

When $[y] = x$ and x is an even integer, then $LLR(y)$ is expressed as:

$$LLR(y) = \pm\log\left(\frac{Pr(b=0|y)}{Pr(b=1|y)}\right) = \pm\log\left(\frac{1-Pr(n<((x-0.5)-y))}{Pr(n<((x-0.5)-y))}\right) \quad (6)$$

where the ± signs are for x being an even/odd integer, respectively.

7 Variation of the Erasure Rate

Erasures are used to better account for symbols where the probability of a bit-error is quite high. For a given channel, by increasing the erasure threshold, the erasure rate is increased and the error rate is decreased - the flip-side is that the rate of correctly mapped symbols also decreases. It is experimentally observed that if the erasure rate is suitably adjusted, the decrease in the rate of correctly mapped symbols is offset by the decrease in the error rate, and the hiding rate is increased.

This method results in increased hiding rates for channels where the effective error rate is high, i.e. it dominates over the erasure probability term. It is experimentally observed that for channels with high error rates (e.g. channels with QF_h of 50-75), using an increased erasure rate results in a higher effective hiding rate. For channels with low error rates (e.g. channels with $QF_h \leq 30$), decreasing the erasure rate increases the hiding rate.

We now show an example of how the effective mapping $P_{c,\hat{c}}$ changes as the erasure cutoff (δ_{dec}) increases. As δ_{dec} is increased from 0.5 (assume that $\delta_{dec} \leq 1$), coefficients in the range $[0.5, \delta_{dec}]$ get mapped to erasures now, and these were mapped to 1 (bit '1' is embedded) when $\delta_{dec} = 0.5$. Thus, $p_{1,1}$ decreases and $p_{1,e}$ increases. Similarly, as the erasure range is increased (from (0.5,0.5) to $(-\delta_{dec}, \delta_{dec})$), $p_{e,e}$ increases while $p_{e,1}$ decreases. From (1), it is seen that $\mathcal{C}_{c,\hat{c}}$ depends on the transition probability matrix $P_{c,\hat{c}}$, whose parameters are expressed in terms of $N_{[a,b]}$, P_n and δ_{dec}. Thus, for a given distribution of the image DCT coefficients and an assumed noise distribution for fixed values of λ, QF_h and QF_a, the capacity can be expressed as a function of δ_{dec}.

Table 6. The average values for $P_{z,\hat{c}}$, $P_{c,\hat{c}}$ and $\mathcal{C}_{c,\hat{c}}$ are computed over 250 images, using B=9, QF_h=75, QF_a=75 and λ=8.

δ_{dec}	$P_{z,\hat{c}}$	$P_{c,\hat{c}}$	$\mathcal{C}_{c,\hat{c}}$
0.5	$\begin{bmatrix} 0.8333 & 0.1673 & 0.0000 \\ 0.0963 & 0.8880 & 0.0157 \\ 0.0000 & 0.0792 & 0.9208 \end{bmatrix}$	$\begin{bmatrix} 0.3557 & 0.1317 & 0.5126 \\ 0.0602 & 0.5503 & 0.3895 \end{bmatrix}$	0.1910
0.6	$\begin{bmatrix} 0.8333 & 0.1673 & 0.0000 \\ 0.0963 & 0.8570 & 0.0467 \\ 0.0000 & 0.0378 & 0.9622 \end{bmatrix}$	$\begin{bmatrix} 0.3557 & 0.1009 & 0.5434 \\ 0.0602 & 0.5258 & 0.4140 \end{bmatrix}$	0.2050
0.7	$\begin{bmatrix} 0.8333 & 0.1673 & 0.0000 \\ 0.0963 & 0.8092 & 0.0944 \\ 0.0000 & 0.0163 & 0.9837 \end{bmatrix}$	$\begin{bmatrix} 0.3557 & 0.0834 & 0.5609 \\ 0.0602 & 0.4967 & 0.4431 \end{bmatrix}$	**0.2074**
0.8	$\begin{bmatrix} 0.8333 & 0.1673 & 0.0000 \\ 0.0963 & 0.7465 & 0.1572 \\ 0.0000 & 0.0061 & 0.9939 \end{bmatrix}$	$\begin{bmatrix} 0.3557 & 0.0745 & 0.5697 \\ 0.0602 & 0.4599 & 0.4799 \end{bmatrix}$	0.1986

We empirically show the best cutoffs to use for different values of QF_h in Fig. 3. The 2×3 mapping between c and z is image-dependent and is unaffected by δ_{dec}. Table 6 shows how the channel capacity varies with the erasure cutoff δ_{dec}, where the embedder side hiding parameters are left unchanged.

8 Results

We show how the effective hiding rate is increased by a combination of the three factors. The %-age improvement in the bpnc is shown for different hiding parameters in Table 7.

For steganalysis, we use a set of 3400 high-quality JPEG images which were originally at a QF of 95 and they were JPEG compressed at a QF of 75 for the experiments. For the experiments, we crop out the central 512×512 region inside each image - the cropping is done for both the cover and stego images.

Steganalysis Feature Used: KF-548 - To improve upon the 274-dimensional calibrated feature [13], Kodovský and Fridrich [9] proposed the use of a 548-dimensional feature set **KF-548** which accounts for both calibrated and un-

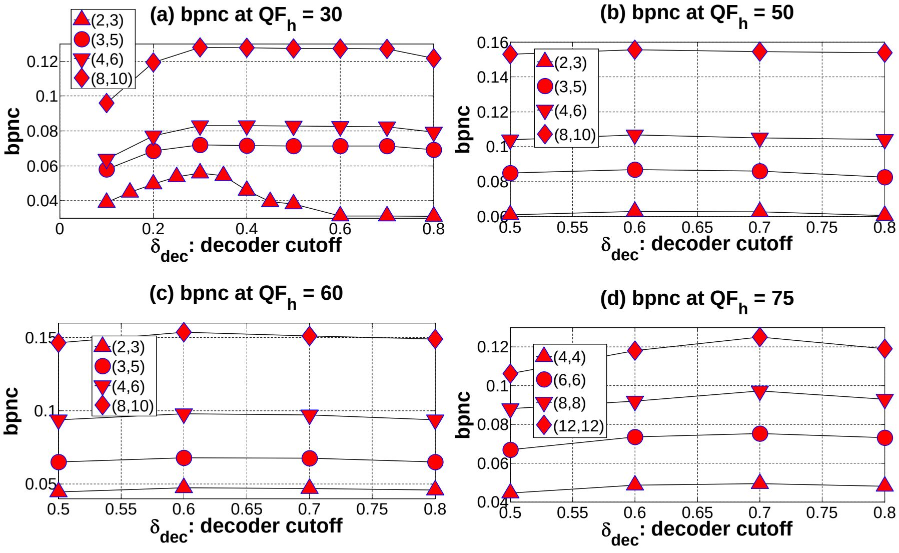

Fig. 3. Variation in bpnc with change in δ_{dec} for different choices of QF_h - the best choices for δ_{dec} are 0.3, 0.6, 0.6 and 0.7 for QF_h of 30, 50, 60 and 75, respectively.

Table 7. The average bpnc results are presented for different hiding conditions. Here, I-LLR refers to the use of image-dependent LLR for decoding. "Puncture" refers to the use of the best combination (λ, λ') for a given λ that maximizes the bpnc. "Erasure" refers to the use of the best choice of δ_{dec}, the decoder cutoff, for a given QF_h, after puncturing. "C-LLR" refers to the use of the coefficient-dependent LLR, after puncturing and using best choice of δ_{dec}. Here, %-gain refers to the fractional gain obtained after using (puncture + erasure + C-LLR), as compared to using only I-LLR. **It is to be emphasized that P_d is unchanged as bpnc is increased for the same hiding parameters by varying the decoder module**.

QF_h	λ	I-LLR	puncture	Erasure	C-LLR	%-gain	QF_h	λ	I-LLR	puncture	Erasure	C-LLR	%-gain
30	2	**0.0318**	0.0425	0.0570	**0.0578**	81.76	50	2	**0.0508**	0.0630	0.0640	**0.0650**	27.95
30	4	**0.0586**	0.0732	0.0833	**0.0846**	44.37	50	4	**0.0932**	0.1040	0.1064	**0.1070**	14.81
30	8	**0.1169**	0.1213	0.1226	**0.1245**	6.50	50	8	**0.1485**	0.1540	0.1555	**0.1580**	6.40
30	12	**0.1387**	0.1401	0.1420	**0.1440**	3.82	50	12	**0.1748**	0.1811	0.1839	**0.1880**	7.55
60	2	**0.0427**	0.0446	0.0476	**0.0519**	21.55	70	2	**0.0230**	0.0230	0.0257	**0.0278**	20.87
60	4	**0.0865**	0.0937	0.0972	**0.1034**	18.38	70	4	**0.0644**	0.0644	0.0701	**0.0768**	19.25
60	8	**0.1427**	0.1464	0.1537	**0.1615**	13.17	70	8	**0.1132**	0.1132	0.1241	**0.1345**	18.82
60	12	**0.1738**	0.1745	0.1858	**0.1957**	12.60	70	12	**0.1379**	0.1379	0.1555	**0.1719**	24.66

calibrated features. Here, the reference feature is used as an additional feature instead of being subtracted from the original feature.

Half of the images are used for training and the other half for testing. We use a support vector machine (SVM) based classifier for steganalysis, where the SVM is trained using the **KF-548** feature. The probability of classifying a test image correctly as cover or stego - the detection accuracy P_d (P_d=50% implies undetectable hiding, and as the detectability improves, P_d increases towards 100%) is obtained using **KF-548**. The steganalysis results are reported in Table 8. Based on the hiding parameters, it is seen that a higher QF_h gives better bpnc-vs-P_d trade-off, i.e. higher bpnc for similar P_d values. E.g. a bpnc of about 0.11 is obtained at a P_d of 0.68 at QF_h=75, while similar P_d values result in bpnc of 0.086 and 0.103 at $QF_h = 50$ and 60, respectively.

Table 8. The steganalysis results are reported using **KF-548**. P_d refers to the detection accuracy. "I-LLR" refers to the bpnc obtained using image-dependent LLRs while "final" refers to the bpnc obtained after using all the three proposed techniques - puncturing, suitably varied erasure cutoff, and coefficient-based LLRs.

Hiding Setup	P_d(%)	I-LLR	final	%-gain
QF_h=50, λ=2	65.00	0.0508	0.0650	27.95
QF_h=50, λ=3	69.00	0.0710	0.0860	21.13
QF_h=60, λ=3	65.80	0.0665	0.0795	19.55
QF_h=60, λ=4	70.06	0.0865	0.1034	19.54
QF_h=75, λ=4	55.00	0.0445	0.0564	26.74
QF_h=75, λ=6	60.40	0.0680	0.0849	24.85
QF_h=75, λ=8	68.00	0.0882	0.1099	24.60

We substituted the DCT-domain hiding with DWT-domain hiding and the same decoding strategies are used for improving the bpnc, as shown in Table 9.

Table 9. The advantages of using the proposed decoder modifications are presented for the DWT domain, in place of the DCT domain, while the YASS based framework of randomly choosing a 8×8 block inside a $B \times B$ big-block is retained.

QF_h	λ	I-LLR	Puncture	Erasure	C-LLR	%-gain	P_d(%)
50	4	**0.0678**	0.0764	0.0803	**0.0848**	25.07	70.99
60	4	**0.0616**	0.0619	0.0656	**0.0710**	15.26	66.93
70	4	**0.0249**	0.0249	0.0286	**0.0348**	39.76	57.75
75	4	**0.0106**	0.0106	0.0117	**0.0169**	59.43	55.56

9 Conclusions

In this paper, we have demonstrated three simple methods to increase the effective data-rate at the decoder side without affecting the embedder process. These methods have been tested on the YASS framework where they have produced 10%-40% improvement in the hiding rate, without affecting the detectability. The methods have been based on the repeat accumulate code based framework, which has been shown to be close to capacity achieving for most hiding conditions. The only hiding condition where the RA code is still somewhat away from being capacity-achieving is for small code-lengths with channels having high error-rates or high erasure rates, and that leaves scope for further improvement. The decoding techniques proposed here are generic and can be incorporated in other steganographic schemes, which involve different methods for hiding coefficient selection, different transform domains for embedding, and different iterative decoding frameworks.

Acknowledgments. This research is supported in part by a grant from ONR # N00014-05-1-0816 and # N00014-10-1-0141.

References

1. Abbasfar, A.: Turbo-like Codes Design for High Speed Decoding. Springer Verlag (2007)
2. Abbasfar, A., Divsalar, D., Yao, K., Inc, R., Los Altos, C.: Accumulate-repeat-accumulate codes. IEEE Transactions on Communications 55(4), 692–702 (2007)
3. Backes, M., Cachin, C.: Public-key steganography with active attacks. In: Theory of Cryptography Conference Proceedings. vol. 3378, pp. 210–226. Springer (2005)
4. Bahl, L., Cocke, J., Jelinek, F., Raviv, J.: Optimal decoding of linear codes for minimizing symbol error rate. IEEE Trans. Inform. Theory 20(2), 284–287 (1974)
5. Chen, B., Wornell, G.W.: Quantization Index Modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. on Info. Theory 47(4), 1423–1443 (May 2001)
6. Divsalar, D., Jin, H., McEliece, R.J.: Coding theorems for turbo-like codes. In: 36th Allerton Conf. on Communications, Control, and Computing. pp. 201–210 (Sep 1998)
7. Kharrazi, M., Sencar, H., Memon, N.: Image steganography: Concepts and practice. Lecture Note Series, Institute for Mathematical Sciences, National University of Singapore (2004)
8. Kodovskỳ, J., Pevnỳ, T., Fridrich, J.: Modern steganalysis can detect YASS. In: Proceedings of SPIE. vol. 7541, p. 754102 (2010)
9. Kodovský, J., Fridrich, J.: Calibration revisited. In: MM & Sec '09: Proceedings of the 11th ACM workshop on Multimedia and security. pp. 63–74. ACM, New York, NY, USA (2009)
10. Li, B., Shi, Y., Huang, J.: Steganalysis of YASS. In: Proceedings of the 10th ACM workshop on Multimedia and security. pp. 139–148. ACM New York, NY, USA (2008)
11. Muller, F.: Distribution shape of two-dimensional DCT coefficients of natural images. Electronics Letters 29(22), 1935–1936 (Oct 1993)
12. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information hiding — A survey. Proceedings of the IEEE, special issue on Identification and Protection of Multimedia Information 87(7), 1062–1078 (1999), `citeseer.nj.nec.com/petitcolas99information.html`
13. Pevný, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. In: Proc. of SPIE. pp. 3 1 – 3 14. San Jose, CA (2007)
14. Reininger, R., Gibson, J.: Distributions of the two-dimensional DCT coefficients for images. Comm., IEEE Trans. on 31(6), 835–839 (Jun 1983)
15. Sarkar, A., Manjunath, B.S.: Image dependent log-likelihood ratio allocation for repeat accumulate code based decoding in data hiding channels. In: Proceedings of SPIE, 2010, Media Forensics and Security. vol. 7541, pp. 754113–754113–6 (Jan 2010), `http://vision.ece.ucsb.edu/publications/LLR_SPIE_2010_anindya.pdf`

16. Shi, Y.Q., Chen, C., Chen, W.: A Markov process based approach to effective attacking JPEG steganography. In: Lecture notes in computer science: 8th International Workshop on Information Hiding. pp. 249–264 (July 2006)
17. Solanki, K., Sarkar, A., Manjunath, B.S.: YASS: Yet Another Steganographic Scheme that resists blind steganalysis. In: 9th International Workshop on Information Hiding. pp. 16–31 (Jun 2007)