# Estimating the Redundancy Factor for RA-encoded sequences and also Studying Steganalysis Performance of YASS

Anindya Sarkar, Upamanyu Madhow and B. S. Manjunath,
Department of Electrical and Computer Engineering,
University of California, Santa Barbara

## 1 Problem Statement

Our recently introduced JPEG steganographic method called *Yet Another Steganographic Scheme* (YASS) can resist blind steganalysis by embedding data in the discrete cosine transform (DCT) domain in randomly chosen image blocks. To maximize the embedding rate for a given image and a specified attack channel, the redundancy factor used by the repeat-accumulate (RA) code based error correction framework in YASS is optimally chosen by the encoder. An efficient method is suggested for the decoder to accurately compute this redundancy factor. We demonstrate the redundancy estimation for the **quantization index modulation** and **matrix embedding** based schemes through Sec. 2-4. The second part of this technical report (Sec. 5) discusses the steganalysis performance of YASS, using different embedding schemes, such as matrix embedding and quantization index modulation, and after using a variety of steganalysis features.

Here, we shall be discussing the estimation of the RA code redundancy factor for the following 2 types of methods. For each method, the databits are RA-encoded using a suitable redundancy factor and then different embedding techniques are used to embed the code bits in the given image.

- QIM-RA: use quantization index modulation (QIM) [1] to embed the RA code bits
- ME-RA: use matrix embedding (ME) to embed the RA code bits

We present a brief introduction into how matrix embedding operates and then also briefly describe YASS, the randomized block-based hiding framework.

**Matrix Embedding Example:**
Consider (7,3) matrix embedding, in which 3 data bits are embedded in 7 host bits. The idea is to perturb the host bits minimally so that they fall in the coset of a linear code, whose syndrome equals the data bits to be hidden. In particular, we consider the (7,4) Hamming code with parity check matrix $\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$. For a host sequence $\mathbf{a} = (1,0,0,1,0,0,1)$, the syndrome $\mathbf{b}'$ is obtained as:

$(\mathbf{b}')^T = \mathbf{H}(\mathbf{a})^T = (0,1,0)^T$, where the operations are performed over the binary field. If the data bits to be embedded are $(0,1,0)$, we are done; we can send the host bits without perturbation. However, suppose that we wish to embed $(0,0,0)$. The aim is to find $\mathbf{\Delta a}$, the perturbation vector for $\mathbf{a}$, with the lowest Hamming weight. Then, $\mathbf{H}(\mathbf{a})^T + \mathbf{H}(\mathbf{\Delta a})^T = (0,0,0)^T$. Therefore, $\mathbf{H}(\mathbf{\Delta a})^T = (0,1,0)^T$. If only the $i^{th}$ element in $\mathbf{\Delta a}$ is 1, then $\mathbf{H}(\mathbf{\Delta a})^T$ equals the $i^{th}$ column in $\mathbf{H}$. The $2^{nd}$ column in $\mathbf{H} = (0,1,0)^T$. Therefore, $\mathbf{\Delta a} = (0,1,0,0,0,0,0)$. Similarly, to embed the data bits $(1,1,1)$, the perturbation $\mathbf{\Delta a}$ is such that $\mathbf{H}(\mathbf{a})^T + \mathbf{H}(\mathbf{\Delta a})^T = (1,1,1)^T \Rightarrow \mathbf{H}(\mathbf{\Delta a})^T = (1,0,1)^T$. Since the $5^{th}$ column of $\mathbf{H} = (1,0,1)^T$, $\mathbf{\Delta a} = (0,0,0,0,1,0,0)$. Similarly, for any three-tuple we might wish to embed, we need to change at most one host bit (Hamming weight of $\mathbf{\Delta a} \leq 1$), which illustrates why matrix embedding is so powerful for passive warden steganography.

**Brief Introduction To YASS:**

The security of YASS [10] can be attributed to the choice of hiding locations. The input image is considered in the pixel domain (it is decompressed if the input is in JPEG format) and then divided into blocks of size $B$×$B$ ($B > 8$), 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, $QF_h$. A band of $\lambda$ AC DCT coefficients lying in the low and mid-frequency range is then used for hiding. After randomized block-based hiding, the resultant image is JPEG compressed at a quality factor of $QF_a$. The embedding rate decreases, as compared to using regular 8×8 blocks, because a lower fraction ($\frac{8\times 8}{B\times B} < 1$) of the DCT coefficients is now considered for embedding. To increase the embedding rate, multiple non-overlapping 8×8 blocks can be fitted in a $B \times B$ block, for $B > 15$. E.g. the number of 8×8 blocks that can be fitted in a 25×25 big-block is $\lfloor 25/8 \rfloor^2 = 9$. The embedding rate is increased, as compared to using $B = 9$, as the effective big-block size $B_{eff}$, which accommodates one 8×8 block, is now reduced from 9 to $\frac{25}{3}$. Once the hiding locations are fixed, the code bits are embedded using ME, while the RA-encoding is done using sufficient redundancy, decided based on the type and severity of the channel attack.

To summarize the roles of the various modules, YASS suggests "where" to embed (randomized block locations), ME/QIM shows "how" to embed (embedding method) and the RA-based ECC framework determines "what" gets embedded (it generates code bits given the data bits) - this is illustrated in Fig. 1.

The work on estimating the redundancy factor ($q$) for RA-encoded sequences for QIM-based YASS has been presented in [7]. Here, we have improved upon the method so that it can be used for ME-based YASS. YASS just provides the stegaongraphic framework and the $q$-estimation strategies depend only on the embedding method (ME/QIM) and not how the hiding locations are chosen. We include the method for $q$-estimation for QIM-RA scheme for completeness and for ease of comparison between the $q$-estimation methods for ME-RA and QIM-RA.

The break-up of the technical report is as follows. Deciding on the best $q$ at the encoder side is discussed in Sec. 2. Estimation of the redundancy factor at the decoder side for QIM-RA is discussed in Sec. 3. The corresponding $q$-estimation for ME-RA is presented in Sec. 4.

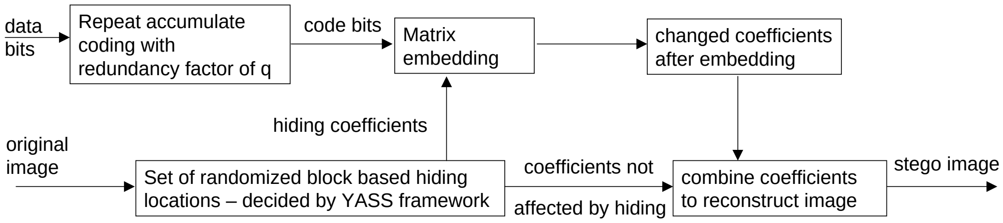
data bits
Repeat accumulate coding with redundancy factor of q
code bits
Matrix embedding
changed coefficients after embedding
hiding coefficients
original image
Set of randomized block based hiding locations – decided by YASS framework
coefficients not affected by hiding
combine coefficients to reconstruct image
stego image

Figure 1: The entire hiding framework using RA coding, matrix embedding (for QIM based framework, this module is substituted by QIM-based embedding) and YASS-based coefficient selection

## 2 Maximizing the Hiding Rate Using Optimum Coding Redundancy at the Encoder and the Decoder

With an increased redundancy factor ($q$) in the RA framework, the hiding rate decreases while the robustness to channel distortions increases. *The hiding rate is maximized if the encoder uses the minimum $q$ that guarantees zero bit error rate (BER) for a given image, a known attack channel and hiding parameters that ensure statistical security*. This redundancy factor is referred to as $q_{opt}$ in subsequent discussions. The decoder knows the embedding method and the error correction code (RA) used, but not the $q$ used at the encoder. We present an efficient method by which the decoder can correctly estimate the $q$ used by the encoder.

The serial concatenated turbo (RA) code based error correction is used in our data hiding setup - Fig. 2 shows the whole framework except for the iterative decoding part at the RA decoder. Let the total number of possible hiding locations in the image be $\ell$. Using a redundancy factor of $q$, the maximum number of embeddable databits, denoted by $N$, equals $\lfloor \ell/q \rfloor$. The encoder repeats the $N$-bit data sequence $u$, as a whole, $q$-times (10), instead of repeating each bit $q$ times. As is shown later, this makes it easier to compute $q$ at the decoder using the auto-correlation (8) of the RA-encoded sequence.

**Steps involved in mapping from $u$ to $y$ at the encoder**

$$[r_{(i-1)N+1} r_{(i-1)N+2} \ldots r_{iN}] = [u_1 u_2 \ldots u_N],\ 1 \leq i \leq q \tag{1}$$

$$x = \pi(r),\ \text{where } \pi \text{ is the interleaver function} \tag{2}$$

$$y_1 = x_1,\ y_n = y_{n-1} \oplus x_n,\ 2 \leq n \leq Nq \tag{3}$$

After data embedding, we get a ternary sequence $z$ of $\{0, 1, e\}$ based on what is actually embedded, where $e$ denotes an erasure (Fig. 2). When a quantized discrete cosine transform (DCT) term in the image lies in the range [-0.5,0.5], an erasure occurs - this maintains perceptual transparency [9]. For DCT terms of higher magnitude, every DCT term is quantized to the nearest odd/even integer to embed 1/0, respectively. The ternary sequence obtained from the hiding locations in the noisy received image, decoded using the same principles used while embedding by the encoder, is called $\hat{y}$.

At the encoder side, the sender transmits a sequence $u$, embeds the RA-encoded sequence $y$ in the image, subjects it to known attacks and finally obtains $\hat{y}$ from the image. Thus, by simulating the exact attack channel, the $2\times3$ transition probability matrix, $p(\hat{y}|y)$ can be computed. The capacity $\mathcal{C}$, for the channel that maps $y$ to $\hat{y}$, is obtained by maximizing the mutual information $I(Y,\hat{Y})$ between the sequences $y$ and $\hat{y}$ (4) - a discrete memoryless channel is assumed here.

$$\mathcal{C} = \max_{p(y)} I(Y,\hat{Y}) = \max_{p(y)} \sum_{y\in\{0,1\}} \sum_{\hat{y}\in\{0,1,e\}} p(y,\hat{y}) \log \left\{ \frac{p(y|\hat{y})}{p(y)} \right\} \tag{4}$$

From a capacity perspective, the minimum redundancy factor needed for perfect data recovery, *assuming an ideal channel code*, is $q_{min} = \lceil \frac{1}{\mathcal{C}} \rceil$. Thus, the minimum possible value of $q_{opt}$ ($q$ needed for perfect data recovery even after channel distortions) for the RA code is $q_{min}$. The sender simulates the decoder and attempts to recover the embedded databits by varying $q$. An upper limit ($q_{max}$) is set on the maximum redundancy factor to be used. Thus, the search for $q_{opt}$, needs to be done in the range $[q_{min}, q_{max}]$ - it will need at most $\log_2(q_{max} - q_{min})$ searches. *It is assumed here that the encoder knows the exact attack*, allowing it to compute $q_{opt}$ precisely. In practice, the range of attacks may be known - the encoder can then design $q_{opt}$ based on the worst-case attack.

In (5) and also later in (7), it is assumed that the output of $\oplus$ is an erasure if any of the input bits is erased.

## 3 Computing the Redundancy Factor for QIM-RA

We discuss the steps involved in estimating the redundancy factor using the $\hat{y}$ sequence at the decoder.

**Steps involved in mapping from $\hat{y}$ to $\hat{r}$ at the decoder**

$$\hat{x}_1 = \hat{y}_1,\ \hat{x}_n = \hat{y}_n \oplus \hat{y}_{n-1},\ 2 \leq n \leq Nq \tag{5}$$

$$\hat{r} = \pi^{-1}(\hat{x}),\ \text{where } \pi^{-1} \text{ is the deinterleaver function} \tag{6}$$

Since the decoder knows the hiding method and assuming that the image size is not altered by the attacks, it can compute $\ell$ - the total number of possible hiding locations. Let the actual $q$ value used by the encoder be $q_{act}$. If the decoder assumes $q$=$q'$, the number of databits equals $\lfloor \ell/q' \rfloor$. In an ideal case, the sequence $\hat{r}$ will be exactly equal to $r$, where $r$ consists of the input message sequence $u$, repeated as a whole. Thus, if $\hat{r}$ is circularly shifted by the assumed input message length $\lfloor \ell/q' \rfloor$, the normalized correlation between the original and the shifted sequences $R_{\hat{r},\hat{r}}(q')$ (8) will be very high if $q'$=$q_{act}$. In (7), $b(q')$ is the sequence obtained after performing element-wise $\oplus$ between the original and shifted sequences, where shift $k = \lfloor \ell/q' \rfloor$. $R_{\hat{r},\hat{r}}(q')$ (8) is the fraction of 0's in $b(q')$ (matches in two corresponding bits after

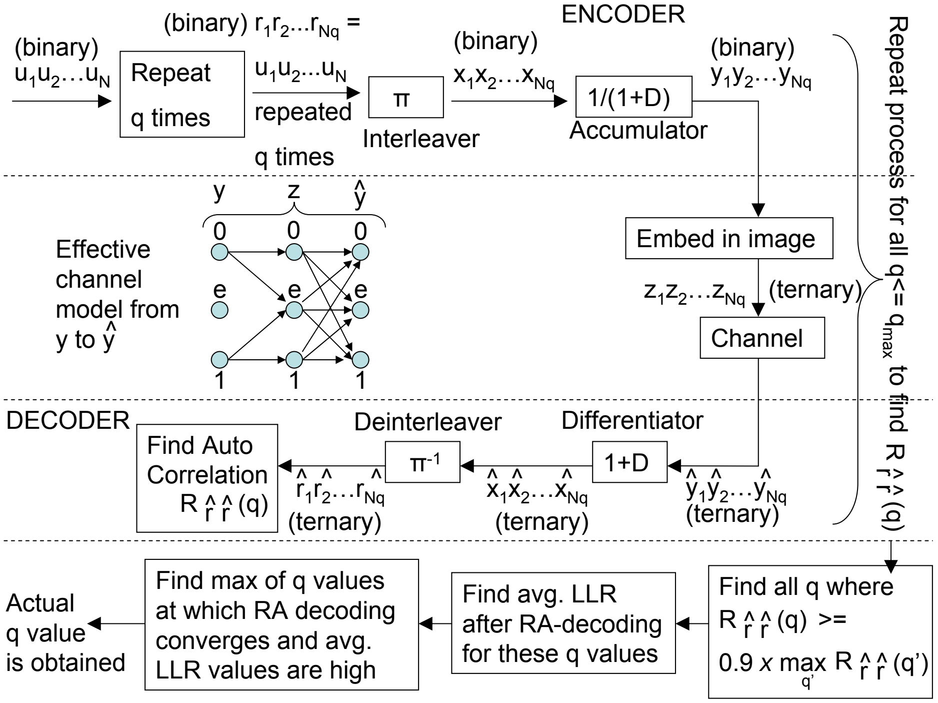
ENCODER
(binary) $u_1u_2 \ldots u_N$
Repeat q times
(binary) $r_1r_2 \ldots r_{Nq}$ = $u_1u_2 \ldots u_N$ repeated q times
π
Interleaver
(binary) $x_1x_2 \ldots x_{Nq}$
1/(1+D)
Accumulator
(binary) $y_1y_2 \ldots y_{Nq}$
Repeat process for all q<= $q_{max}$ to find $R_{\hat{r}\hat{r}}(q)$
y z $\hat{y}$
0 0 0
e e e
1 1 1
Effective channel model from y to $\hat{y}$
Embed in image
$z_1z_2 \ldots z_{Nq}$ (ternary)
Channel
DECODER
Find Auto Correlation $R_{\hat{r}\hat{r}}(q)$
Deinterleaver
$\pi^{-1}$
$\hat{r}_1\hat{r}_2 \ldots \hat{r}_{Nq}$ (ternary)
Differentiator
1+D
$\hat{x}_1\hat{x}_2 \ldots \hat{x}_{Nq}$ (ternary)
$\hat{y}_1\hat{y}_2 \ldots \hat{y}_{Nq}$ (ternary)
Find all q where $R_{\hat{r}\hat{r}}(q)$ >= 0.9 x $\max_{q'} R_{\hat{r}\hat{r}}(q')$
Find avg. LLR after RA-decoding for these q values
Find max of q values at which RA decoding converges and avg. LLR values are high
Actual q value is obtained

Figure 2: The data hiding system using RA-code based error correction, where $q$ is efficiently estimated at the decoder

$\oplus$ result in 0's), without considering the erasures.

$$b(q') = (\{\hat{r}_1 \ldots \hat{r}_{kq'}\} \oplus \{\hat{r}_{kq'-k+1} \ldots \hat{r}_{kq'}\hat{r}_1 \ldots \hat{r}_{kq'-k}\}) \tag{7}$$

$$\text{and shift } k = \lfloor \ell/q' \rfloor \text{ is the assumed number of databits}$$

$$R_{\hat{r},\hat{r}}(q') = \frac{\text{Number of 0's in } b(q')}{\text{Number of 0's and 1's in } b(q')} \tag{8}$$

$$\mathcal{Q}_{top} = \left\{ q' : R_{\hat{r},\hat{r}}(q') >= 0.9 \times ( \max_{q' \leq q_{max}} R_{\hat{r},\hat{r}}(q')) \right\} \tag{9}$$

The correlation is also high when the shift equals a multiple of the actual message length, i.e. $q'=q_{act}/m,\ m \in \mathbb{Z}^+$. Apart from the correlation peaks at $q_{act}$ and its sub-multiples, other peaks may occur due to errors and erasures. In the experiments, the set of $q$ values, $\mathcal{Q}_{top}$ (16), at which the correlation exceeds 90% of the maximum $R_{\hat{r},\hat{r}}$ value, are selected - the 90% cutoff was empirically determined. The turbo decoder is then run for these $q$ values and the log-likelihood ratios (LLR) are computed for the extracted databits in each case. It is seen that due to a noisy channel, decoding may converge (two consecutive iterations produce the same output sequence) at values other than $q_{act}/m,\ m \in \mathbb{Z}^+$. However, the LLR value, averaged over the databits, is high only when perfect decoding occurs. It is seen that the maximum average LLR values occur only at $q_{act}$ and its sub-multiples. Thus, the solution is to consider the maximum of these $q$ values as $q_{act}$, as shown in Fig. 2. This method of estimating $q$ for RA encoding is found to work even at high erasure rates ($\geq 95\%$).

**Observations about the $q$-estimation method**

- The use of auto-correlation based peaks reduces the search space for $q$ while the average LLR-based measure, followed by taking the maximum, helps to identify the actual $q$.
- For our experiments, the search range for $q$ was $[2, 50]$.
- Though the correlation in $\hat{r}$ is used for $q$-estimation, this correlation is not detectable by an adversary; $\hat{r}$ is obtained from $\hat{y}$ only after applying the deinterleaver ($\pi^{-1}$) - the key to generate $\pi^{-1}$ is not known to an adversary.
- The $q$-estimation method is generic enough to be used for any hiding scheme which uses RA-$q$ based error correction.

## 4 Estimating the Redundancy Factor for ME-RA

The proposed $q$-estimation framework for ME-RA is a modification of the approach used for the QIM-RA scheme.

**System Framework:**

Fig. 3 shows the end-to-end hiding framework except for the iterative decoding part at the RA decoder. If there are $\ell$ possible hiding locations, and (7,3) ME is used, the actual number of coded bits that can be embedded $\ell' = \lfloor \ell \times 3/7 \rfloor$. For a redundancy factor of $q$, the number of data bits for (7,3) ME, $N' = \lfloor \ell'/q \rfloor$. On inputting $\{u_n\}_{n=1}^{N'}$, the sequence of $N'$ data bits to a $q$-times repeater block, the output is $\{r_n\}_{n=1}^{N'q}$ (10), which is then passed through the interleaver $\pi$. The resulting sequence is $\{x_n\}_{n=1}^{N'q}$ (11), which is then passed through the accumulator ($\frac{1}{1+D}$) to produce $\{c_n\}_{n=1}^{N'q}$ (12), the encoded output.

**Steps involved in mapping from $\{u_n\}$ to $\{c_n\}$ at the encoder**

$$[r_{(i-1)N'+1} r_{(i-1)N'+2} \ldots r_{iN'}] = [u_1 u_2 \ldots u_{N'}],\ 1 \leq i \leq q \tag{10}$$

$$\{x_1, x_2, \cdots, x_{N'q}\} = \pi(\{r_1, r_2, \cdots, r_{N'q}\}),\ \text{where } \pi \text{ is the interleaver function} \tag{11}$$

$$c_1 = x_1,\ c_n = c_{n-1} \oplus x_n,\ 2 \leq n \leq N'q \tag{12}$$

The RA-encoded sequence $\{c_n\}$ is embedded in the image using matrix embedding. After embedding, we get a ternary sequence $\{z_n\}$ of $\{0, 1, e\}$ based on what is actually embedded, where $e$ denotes an erasure (Fig. 3). The sequence of LLR values obtained from the hiding locations in the noisy received image is called $\{\hat{c}_n\}$.

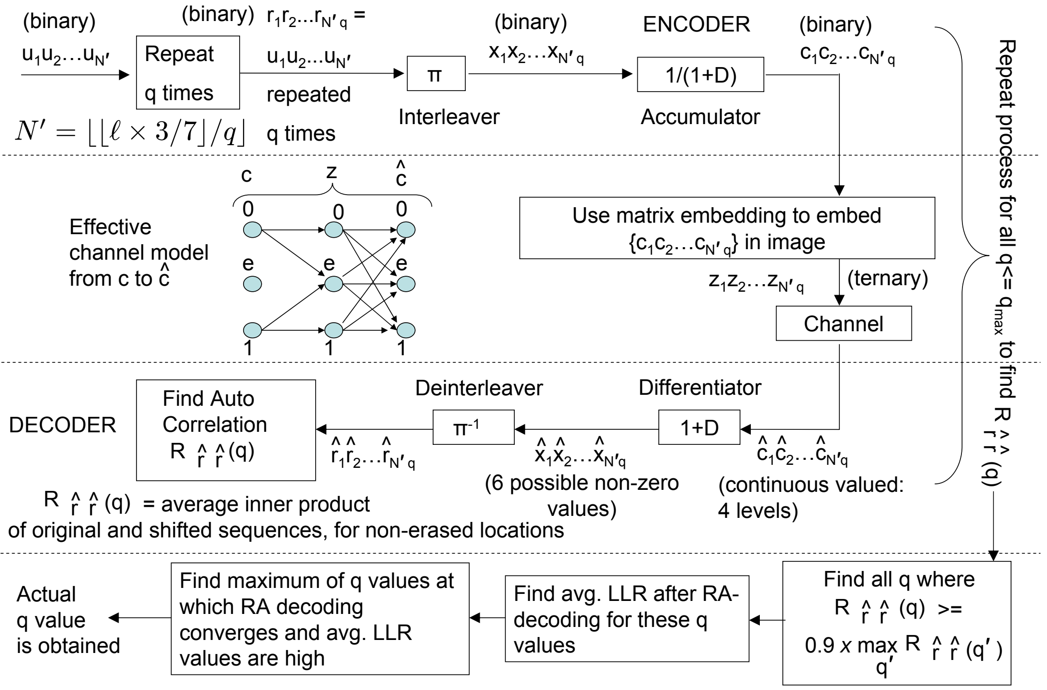

Figure 3: The data hiding setup for ME-RA method where the decoder has to correctly estimate the $q$ that was independently decided upon by the encoder.

**Decoder Outputs:**

The LLR values, computed for ME-RA using M1, M2 or M3, are continuous valued. However, for providing the sequence of the LLR values $\{\hat{c}_n\}$ as input to the differentiator $(1+D)$, the LLR values have to be quantized into a finite number of levels. The values in $\{\hat{c}_n\}$ are split into 4 zones ($[-\infty,-\delta)$, $[-\delta,0)$, $(0,\delta]$, $(\delta,\infty]$), based on a suitably chosen threshold $\delta$. Here, instead of a binary output for the $(1+D)$ block, the elements in $\{\hat{x}_n\}$ are assumed to have positive (instead of 1) and negative (instead of 0) values. The magnitude of the $\{\hat{x}_n\}$ terms is quantized to 3 levels $\{\mathcal{L},\mathcal{M},\mathcal{H}\}$, where $\mathcal{L}<\mathcal{M}<\mathcal{H}$. These values indicate the confidence we have in the decoded $\{\hat{x}_n\}$ values based on the LLR terms $\{\hat{c}_n\}$. E.g. if both $\hat{c}_n$ and $\hat{c}_{n-1}$ are positive, then $\hat{x}_n$ should be negative ($1\oplus 1=0$, here "positive" ⇔ 1 and "negative" ⇔ 0). If both $\hat{c}_n$ and $\hat{c}_{n-1}$ exceed $\delta$, we have high confidence in both these terms being positive and $\hat{x}_n$ being negative and hence, $\hat{x}_n=-\mathcal{H}$. The various possibilities are covered in (13), where we show how the terms in $\{\hat{x}_n\}$ are computed from $\{\hat{c}_n\}$.

$$\begin{array}{rlrl} \hat{x}_n & > 0 \text{ if } \hat{c}_n \times \hat{c}_{n-1} < 0, & |\hat{x}_n| = \mathcal{L} & \text{if } |\hat{c}_n| < \delta, \text{ and } |\hat{c}_{n-1}| < \delta \\ & < 0 \text{ if } \hat{c}_n \times \hat{c}_{n-1} > 0, & = \mathcal{M} & \text{if } |\hat{c}_n| \geq \delta \text{ and } |\hat{c}_{n-1}| < \delta, \text{ or vice versa} \\ & = 0 \text{ if } \hat{c}_n \times \hat{c}_{n-1} = 0, & = \mathcal{H} & \text{if } |\hat{c}_n| \geq \delta, \text{ and } |\hat{c}_{n-1}| \geq \delta \end{array} \qquad (13)$$

**Correlation Computation:**

The next issue is computing the correlation function $R_{\hat{r},\hat{r}}(q')$, where the sequence $\{\hat{r}_n\}$ is obtained after deinterleaving $\{\hat{x}_n\}$, where $\{\hat{r}_n\} = \pi^{-1}(\{\hat{x}_n\})$. The correlation between $\{\hat{r}_n\}$ and its shifted sequence is computed as a normalized inner-product. For an assumed redundancy factor of $q'$, we perform element-by-element multiplication between the 2 sequences, $\{\hat{r}_1 \ldots \hat{r}_{kq'}\}$ and $\{\hat{r}_{kq'-k+1} \ldots \hat{r}_{kq'} \hat{r}_1 \ldots \hat{r}_{kq'-k}\}$, (shift $k = \lfloor \ell'/q' \rfloor$ is the assumed number of data bits) to obtain the sequence $\{s_{q'}\}$ (14) and then the average is taken over the non-zero elements in $\{s_{q'}\}$ to compute $R_{\hat{r},\hat{r}}(q')$ (15). The zeroes in $\{s_{q'}\}$ correspond to those locations where at least one of the corresponding elements in the original and shifted $\{\hat{r}_n\}$ sequences are erased. Thus, a main difference between the $q$-estimation strategies for QIM-RA and ME-RA is that the correlation used for QIM-RA is an inner-product between binary sequences while for ME-RA, it is an inner-product between continuous valued sequences.

$$s_{q',i} = \hat{r}_i \times \hat{r}_{kq'-k+i}, \; 1 \leq i \leq kq', \text{ where shift } k = \lfloor \ell'/q' \rfloor \text{ is the assumed number of data bits} \qquad (14)$$

$$R_{\hat{r},\hat{r}}(q') = (\sum_i s_{q',i})/(\text{number of non-zeros in } \{s_{q'}\}) \qquad (15)$$

$$\mathcal{Q}_{top} = \left\{ q' : R_{\hat{r},\hat{r}}(q') >= 0.9 \times (\max_{q_1 \in \{q\}} R_{\hat{r},\hat{r}}(q_1)) \right\} \qquad (16)$$

where $\{q\} = \{1, 2, \cdots, q_{max}\}$ is the set of possible $q$-values, assuming a maximum $q$ of $q_{max}$.

Let the actual $q$ value used by RA coding equal $q_{act}$. In a noise-free scenario, the $\hat{r}_n$ values will be high or low depending on whether the corresponding values in $\{r_n\}$ are 1 or 0. The correlation $R_{\hat{r},\hat{r}}(q')$ is high when the shift ($k = \lfloor \ell'/q' \rfloor$) equals a multiple of the actual message length, i.e. $q'$=$q_{act}/m, \; m \in \mathbb{Z}^+$. Hence, we can expect peaks at $q_{act}$ and/or its sub-multiples. In practice, due to errors and erasures, peaks can occur at other $q$ values. *Hence, the correlation is used to prune the search space for $q_{act}$ but is not the only deciding criterion.*

**Selecting Right Value for Redundancy Factor:**

The method for $q$-estimation from the top correlation values is very similar to the method used for QIM-RA. In the experiments, $\mathcal{Q}_{top}$ (16), the set of $q$ values at which the correlation exceeds 90% of the maximum $R_{\hat{r},\hat{r}}$ value, is considered as the set of possible $q_{act}$ values - the 90% cutoff was empirically determined. The turbo decoder is then run for the $q$ values in $\mathcal{Q}_{top}$ and the LLR values are computed for the extracted data bits in each case. It is seen that for noisy channels, decoding may converge (two consecutive iterations produce the same output sequence) at values other than $q_{act}/m, \; m \in \mathbb{Z}^+$. *However, the LLR value, averaged over the* *data bits, is high only when perfect decoding occurs*. Hence, the maximum average LLR values occur only at $q_{act}$ and its sub-multiples. Thus, the solution is to consider the maximum of these $q$ values (corresponding to high LLR) as $q_{act}$, as shown in Fig. 3. Running the RA decoder for all $q \in \{q\}$ involves a much higher level of computational complexity than finding $R_{\hat{r},\hat{r}}$ for all $q$, selecting the topmost correlation values, and running the RA decoder on the pruned set of $q$ values. This method of estimating $q$ is found to work even at high erasure rates.

**Performance Comparison:**
To compare the relative performance of $q$-estimation, based on *only correlation*, between QIM-RA and ME-RA, we use the separation in the correlation peaks as an index. We also find the number of times there is an error in $q$-estimation using just the correlation (the $q$ corresponding to the maximum correlation value is chosen as $q_{act}$). To reiterate, these errors are rectified once the final $q$ estimation is done based on the maximum average LLR values.

Let $\mathcal{A}$ and $\mathcal{B}$ denote the two sets - $\{q_{act}/m,\ m \in \mathbb{Z}^+\}$ and $\{\{q\} \setminus \{q_{act}/m\}_{m \in \mathbb{Z}^+}\}$, corresponding to "correct" and "wrong" choice of $q$ values, respectively. When the maximum correlation value comes from an element in $\mathcal{A}$, we classify it as "correct"; otherwise, it is an error. Also, to quantify the discriminability between elements in $\mathcal{A}$ and $\mathcal{B}$ provided by the correlation based approach, we compute the difference, $R_{diff}$ (17), between the topmost correlation values among elements in $\mathcal{A}$ and $\mathcal{B}$, for both the "correct" and "wrong" cases.

$$R_{diff} = \max_{q' \in \mathcal{A}} R_{\hat{r},\hat{r}}(q') - \max_{q' \in \mathcal{B}} R_{\hat{r},\hat{r}}(q') \tag{17}$$

For the correct/wrong cases, we would want the value of $R_{diff}$ to be higher/lower, respectively.

We use the (7,3) ME-RA scheme, with M3 based decoding, for the $q$-estimation experiments. Suitable values for $\delta, \mathcal{L}, \mathcal{M}$ and $\mathcal{H}$ are empirically determined: $\delta = 0.2, \mathcal{L} = 0.10, \mathcal{M} = 0.75$ and $\mathcal{H} = 1.90$. The results are shown for 3 cases - $QF_h$ is varied from 50 to 70 and $QF_a$ is set to 75 (Table 1). Table 1 shows that ME-RA performs better than QIM-RA, both in terms of having lesser errors and in having better separation in the correlation peaks, between elements in $\mathcal{A}$ and $\mathcal{B}$.

# 5 Steganalysis Experiments and Results

The focus of these experiments are to demonstrate the following:
(i) We first compare the detectability of both the QIM-RA and the ME-RA-puncture schemes against steganalysis, at similar hiding rates (shown later in Tables 2 and 3). The hiding rates are adjusted by varying $B$ and the number of AC DCT coefficients used for hiding ($\lambda$).
(ii) We also study the detection performance when hiding is performed in a randomly selected set of AC DCT coefficients instead of always choosing the top $\lambda$ coefficients (as returned by zigzag scan). This is demonstrated later through Tables 4 and 5.
(iii) We also investigate *the level of noise attacks upto which ME performs better than QIM*, as shown later in Table 6.

Table 1: The results of estimating $q$ over 50 images, where the $q$ used for RA coding is varied from 10-43, are presented here, for (7,3) ME-RA and QIM-RA methods. Thus, the total number of cases over which $q$ is estimated = $50 \times 34 = 1700$. The big-block size $B$ is set to 9, while $QF_a$=75. An "error" occurs when the top peak in the correlation based method does not correspond to the actual $q$ or its sub-multiples. Based on just the correlation, ME-RA performs better than QIM-RA in $q$-estimation.

| $QF_h$ | Method | error fraction | avg. $R_{diff}$ (correct cases) | avg. $R_{diff}$ (wrong cases) |
|---|---|---|---|---|
| 50 | QIM-RA | 50/1700 | 0.1500 | -0.2972 |
| | ME-RA | 23/1700 | 1.0431 | -0.1835 |
| 60 | QIM-RA | 151/1700 | 0.0775 | -0.2664 |
| | ME-RA | 91/1700 | 0.5272 | -0.2111 |
| 70 | QIM-RA | 393/1700 | 0.0198 | -0.3083 |
| | ME-RA | 364/1700 | 0.1339 | -0.1663 |

(iv) We also present the steganalysis results using some recently proposed features, most of which were designed specifically to detect YASS (Table 7 and Table 8).
(v) We also observe how the steganalysis performance is improved on using a larger sized dataset for training, as shown in Table 9.

## 5.1 Setup for Steganalysis Experiments

The experiments are done on a set of 1630 high-quality JPEG images taken with a Canon S2-IS Powershot camera; the images were originally at a QF of 95 and they were JPEG compressed at a QF of 75 for the experiments [1]. The advertised QF ($QF_a$) is therefore kept at 75, so that both the cover and stego images, considered for steganalysis, are at the same JPEG QF.

**Steganalysis Performance Measures:** The steganalysis results are expressed in terms of the detection probability $P_{detect}$ (18) while the embedding rates are expressed in terms of the *bpnc*. We train a support vector machine (SVM) on a set of known stego and cover images. The SVM classifier has to distinguish between two classes of images: cover (class '0') and stego (class '1'). Let $X_0$ and $X_1$ denote the events that the image being observed belongs to classes '0' and '1', respectively. On the detection side, let $Y_0$ and $Y_1$ denote the events that the observed image is classified as belonging to classes '0' and '1', respectively. The probability of detection, $P_{detect}$, is defined as follows:

$$\begin{aligned} P_{error} &= P(X_0)P(Y_1|X_0) + P(X_1)P(Y_0|X_1) = \frac{1}{2}P_{FA} + \frac{1}{2}P_{miss}, \text{ for } P(X_0) = P(X_1) = \frac{1}{2} \\ P_{detect} &= 1 - P_{error} \end{aligned} \quad (18)$$

[1] We have experimentally observed that the detectability is higher using high quality JPEG images than images taken with the same camera, but at poorer quality, i.e. JPEG compressed with lower QF. Hence, we use high-quality images for our experimental setup to show that ME-based YASS is more undetectable as compared to QIM-based YASS.

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

**Features Used for Steganalysis:** The following features are used for steganalysis as these have generally been reported as having the best detection performance among modern JPEG steganalyzers.

1. **PF-219/324/274**: Pevny and Fridrich's 274-dim feature vector (**PF-274**) is based on the self-calibration method [6] and it merges Markov and DCT features. The extended DCT feature set and Markov features are 193-dim (**PF-193**) and 81-dim, respectively. The logic behind the fusion is that while Markov features capture the intra-block dependency among DCT coefficients of similar spatial frequencies, the DCT features capture the inter-block dependencies. For the extended DCT features [6, 3], the authors have a 219-dim implementation (**PF-219**) [2]. The Markov features (**PF-324**) are obtained based on the 324-dim intra-block correlation based feature set (**Shi-324**) proposed by Shi et al [8] - the only difference being that the features are "calibrated" in [6].

2. **Chen-486**: Another steganalysis scheme that accounts for both intra and inter-block correlation among JPEG DCT coefficients is the 486-dim feature vector, proposed by Chen et al [2]. It improves upon the 324-dim intra-block correlation based feature [8].

## 5.2 Discussion of Experimental Results

**Comparison after Varying YASS Big-block Size $B$:** The detection performance, in terms of $P_{detect}$ (18), and the embedding rate, in terms of bpnc, are compared for QIM-RA and "ME-RA-puncture", using big-block size $B$ = 9 and 10 (Table 2), and 25 and 49 (Table 3). The ME based method has been experimented with for both the (7,3) and (3,2) encoding schemes. "QIM-RA: $n$ terms" refers to that QIM-RA (QIM-based YASS where RA-coding is used as ECC) scheme where the first $n$ AC DCT elements encountered during zigzag scan per 8×8 block are used for embedding, i.e. the size of the embedding band per 8×8 block $\lambda = n$.

From these tables, it is seen that $P_{detect}$ is comparable for "QIM-RA: 2 terms" and "ME-RA-puncture (7,3)" while the latter has a higher embedding rate. The bpnc for "ME-RA-puncture (7,3)" (or ME-RA-puncture (3,2)) is higher than that of "QIM-RA: 4 terms" (or QIM-RA: 6 terms) while the latter is more detectable, for the self-calibration based features. It is seen that YASS is more detectable using the self-calibration based features, than using Chen-486. Hence, the performance improvement of ME over QIM (lower $P_{detect}$ at similar bpnc values) is more significant for PF-219/324/274 features.

[2] **PF-219** differs from **PF-193** in the following ways: (i) in **PF-219**, there are 25 co-occurrence features for both the horizontal and vertical directions - these are averaged to give 25 features in **PF-193**. (ii) Instead of 1 variation feature in **PF-193**, there are 2 variation features (for horizontal and vertical directions, separately) in **PF-219**.

Table 2: Comparing detection performance ($P_{detect}$) and embedding rate (bpnc) using QIM-RA and “ME-RA-puncture” schemes - for $B$=9 and 10, $QF_h$=50, $QF_a$=75. **The bpnc for “ME-RA-puncture (7,3)” (or ME-RA-puncture (3,2)) is higher than that of “QIM-RA: 4 terms” (or QIM-RA: 6 terms) while the latter is more detectable, for the self-calibration based features**.

| Hiding | big-block size $B$=9 | | | | | big-block size $B$=10 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Scheme | PF-219 | PF-324 | PF-274 | Chen-486 | bpnc | PF-219 | PF-324 | PF-274 | Chen-486 | bpnc |
| QIM-RA: 2 terms | 69.45 | 65.52 | 67.73 | 52.39 | 0.0493 | 68.83 | 65.28 | 67.85 | 52.52 | 0.0382 |
| QIM-RA: 4 terms | 80.00 | 77.18 | 79.39 | 56.20 | 0.0864 | 78.53 | 74.97 | 77.91 | 55.09 | 0.0700 |
| QIM-RA: 6 terms | 81.84 | 77.67 | 84.05 | 57.55 | 0.1138 | 78.90 | 79.26 | 79.39 | 55.95 | 0.0923 |
| ME-RA-puncture (7,3) | 64.79 | 65.40 | 69.45 | 55.95 | **0.0975** | 63.31 | 68.83 | 67.61 | 54.36 | **0.0805** |
| ME-RA-puncture (3,2) | 74.97 | 72.27 | 78.65 | 61.60 | **0.1200** | 73.87 | 78.77 | 78.77 | 59.02 | **0.0998** |

Table 3: Comparing $P_{detect}$ and bpnc for **QIM-RA** and **“ME-RA-puncture”** - for $B$=25 and 49, $QF_h$=50, $QF_a$=75

| Hiding | big-block size $B = 25$ | | | | | big-block size $B = 49$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Scheme | PF-219 | PF-324 | PF-274 | Chen-486 | bpnc | PF-219 | PF-324 | PF-274 | Chen-486 | bpnc |
| QIM-RA: 2 terms | 71.53 | 66.38 | 71.17 | 53.74 | 0.0588 | 71.17 | 68.96 | 71.78 | 54.23 | 0.0606 |
| QIM-RA: 4 terms | 82.70 | 79.26 | 82.45 | 57.55 | 0.1018 | 82.33 | 80.00 | 83.56 | 59.14 | 0.1048 |
| QIM-RA: 6 terms | 84.29 | 80.61 | 86.26 | 59.51 | 0.1336 | 87.98 | 84.54 | 88.34 | 61.47 | 0.1379 |
| ME-RA-puncture (7,3) | 72.15 | 73.99 | 75.95 | 59.14 | **0.1106** | 69.08 | 67.61 | 71.04 | 56.69 | **0.1136** |
| ME-RA-puncture (3,2) | 77.30 | 82.82 | 81.35 | 62.54 | **0.1382** | 82.94 | 84.91 | 84.91 | 63.80 | **0.1421** |

Depending on the bpnc requirements for a certain stego scheme, one can decide whether to use (3,2) or (7,3) matrix embedding - the former allows for higher bpnc while the latter is more undetectable. Using (15,4) code for ME results in very low hiding rates and hence has not been considered.

**Comparison after Further Randomization for QIM-RA:** In the experiments discussed above, the top AC DCT elements, encountered after zigzag scan, are used for embedding. While the top DCT elements are generally higher in magnitude (non-erasure locations) making them suitable for embedding, most detection schemes (e.g. PF-219/274) focus on these coefficients - hence, using these coefficients for hiding increases the hiding rate and also helps in detection. *To make detection more difficult, we choose a certain number of DCT terms randomly out of the top 19 coefficients for the “QIM-RA: rand-$n$” scheme.* In Tables 4 and 5, “QIM-RA: rand-$n$” refers to the QIM-RA scheme, where $n$ randomly chosen AC DCT terms out of the top 19 are used for embedding. For this scheme, we vary $n$, the number of DCT coefficients in the embedding band, to make the hiding rate comparable to that resulting from the ME based scheme - the detection rates of the QIM and ME based methods are then compared. We experiment with hiding at $QF_h$ = 50, 60 and 70, as shown in Tables 4 and 5.

From Table 4, “ME-RA-puncture(7,3)” (or ME-RA-puncture(3,2)) performs better than “QIM-RA:

Table 4: Comparing $P_{detect}$ and bpnc for QIM-RA and "ME-RA-puncture", for $B$=9 and 10, $QF_h = 50$, and using randomly chosen DCT coefficients for QIM-RA. **"ME-RA-puncture(7,3)" performs better than "QIM-RA: rand-8/10" while "ME-RA-puncture(3,2)" performs better than "QIM-RA: rand-12"**.

| Hiding | $B = 9, QF_h = 50$ | | | | | $B = 10, QF_h = 50$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Scheme | PF-219 | PF-324 | PF-274 | Chen-486 | bpnc | PF-219 | PF-324 | PF-274 | Chen-486 | bpnc |
| QIM-RA: rand-8 | 69.45 | 66.99 | 70.43 | 53.37 | 0.0700 | 68.83 | 68.10 | 69.82 | 52.52 | 0.0530 |
| QIM-RA: rand-10 | 71.04 | 73.13 | 74.36 | 55.83 | 0.0850 | 78.16 | 78.28 | 80.25 | 55.34 | 0.0700 |
| QIM-RA: rand-12 | 78.41 | 76.81 | 80.61 | 57.30 | 0.1115 | 81.23 | 80.98 | 83.56 | 56.07 | 0.0880 |
| ME-RA-puncture (7,3) | 64.79 | 65.40 | 69.45 | 55.95 | **0.0975** | 63.31 | 68.83 | 67.61 | 54.36 | **0.0805** |
| ME-RA-puncture (3,2) | 74.97 | 72.27 | 78.65 | 61.60 | **0.1200** | 73.87 | 78.77 | 78.77 | 59.02 | **0.0998** |

Table 5: Comparing $P_{detect}$ and bpnc for QIM-RA and "ME-RA-puncture", using $B$=9 and $QF_h$=60 and 70, and using randomly chosen DCT coefficients for QIM-RA. **"ME-RA-puncture(7,3)" performs better than "QIM-RA: rand-8" while "ME-RA-puncture(3,2)" performs better than "QIM-RA: rand-10/12"**.

| Hiding | $B = 9, QF_h = 60$ | | | | | $B = 9, QF_h = 70$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Scheme | PF-219 | PF-324 | PF-274 | Chen-486 | bpnc | PF-219 | PF-324 | PF-274 | Chen-486 | bpnc |
| QIM-RA: rand-8 | 74.72 | 73.13 | 74.72 | 53.99 | 0.0760 | 61.23 | 62.58 | 63.19 | 52.39 | 0.0430 |
| QIM-RA: rand-10 | 77.18 | 79.63 | 80.00 | 55.95 | 0.0913 | 64.79 | 65.64 | 66.50 | 53.37 | 0.0550 |
| QIM-RA: rand-12 | 79.63 | 85.03 | 85.77 | 64.79 | 0.1094 | 69.69 | 69.33 | 70.43 | 55.21 | 0.0720 |
| ME-RA-puncture (7,3) | 63.34 | 64.61 | 66.27 | 55.09 | **0.0916** | 56.81 | 59.75 | 57.18 | 52.39 | **0.0537** |
| ME-RA-puncture (3,2) | 73.55 | 71.81 | 75.12 | 62.82 | **0.1161** | 63.31 | 68.34 | 65.03 | 54.85 | **0.0780** |

Table 6: Comparing bpnc under various attacks - **"QIM-$n$"** refers to the **"QIM-RA: $n$ terms"** method, while **(p,q)** refers to the **"ME-RA-puncture (p,q)"** method. For hiding, we use $QF_h = 50$, $B = 9$, and after the attack, the images are JPEG compressed using $QF_a = 75$. Here, the bpnc for "ME-RA-puncture(7,3)" and "ME-RA-puncture(3,2)" are compared with that of QIM-4 and QIM-6, respectively.

| Gamma correction: $\gamma < 1$ | | | | | Gamma correction: $\gamma > 1$ | | | | | AWGN attack: SNR (dB) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\gamma$ | QIM-4 | QIM-6 | (7,3) | (3,2) | $\gamma$ | QIM-4 | QIM-6 | (7,3) | (3,2) | SNR | QIM-4 | QIM-6 | (7,3) | (3,2) |
| **0.99** | 0.0851 | 0.1125 | 0.0953 | 0.1191 | **1.01** | 0.0854 | 0.1125 | 0.0959 | 0.1194 | **50** | 0.0860 | 0.1133 | 0.0952 | 0.1190 |
| **0.98** | 0.0839 | 0.1105 | 0.0929 | 0.1171 | **1.02** | 0.0843 | 0.1114 | 0.0935 | 0.1171 | **45** | 0.0859 | 0.1125 | 0.0940 | 0.1179 |
| **0.95** | 0.0783 | 0.1013 | 0.0849 | 0.1069 | **1.05** | 0.0799 | 0.1026 | 0.0863 | 0.1095 | **40** | 0.0840 | 0.1096 | 0.0885 | 0.1126 |
| **0.90** | 0.0608 | 0.0799 | 0.0655 | 0.0845 | **1.10** | 0.0631 | 0.0827 | 0.0685 | 0.0893 | **35** | 0.0728 | 0.0912 | 0.0659 | 0.0889 |
| **0.80** | 0.0307 | 0.0401 | 0.0200 | 0.0250 | **1.20** | 0.0366 | 0.0465 | 0.0260 | 0.0400 | **30** | 0.0393 | 0.0485 | 0.0200 | 0.0260 |

rand-8/10" (or QIM-RA: rand-12) - by having higher bpnc for similar $P_{detect}$, at $QF_h$ of 50. From Table 5, "ME-RA-puncture(7,3)" performs better than "QIM-RA: rand-8" while "ME-RA-puncture(3,2)" performs better than "QIM-RA: rand-10/12", at $QF_h$ of 60 and 70.

**Robustness Comparison for Various Noise Attacks:** We now study how the bpnc is affected by additional noise attacks for these schemes. The YASS framework can be made robust against various global (*and not local*) attacks by adjusting the RA-code redundancy factor. We consider a wider range of attacks - gamma variation and additive white Gaussian noise (AWGN) attacks, which are followed by JPEG compression at $QF_a$=75. It is seen that for higher noise levels, ($|\gamma - 1| > 0.10$, for gamma variation, or SNR $\leq 35$ dB, for AWGN) the bpnc is significantly lower for the ME based method, as compared to QIM-RA, for similar detection rates (Table 6).

**Using Recent Steganalysis Features more tuned to detect YASS:** We explain the following features and then show the steganalysis performance using these features in Tables 7 and 8:
(i) **KF-548**: To improve upon the **PF-274** feature, Kodovsky and Fridrich [4] proposed the use of a 548-dimensional feature set which accounts for both calibrated and uncalibrated features. Here, the reference feature is used as an additional feature instead of being subtracted from the original feature.
(ii) **Li-14** and **Li-2**: In [5], Li et al propose the use of the frequency of re-quantized DCT coefficients in the candidate embedding band which round off to zero. The $(2i-1)^{th}$ and $(2i)^{th}$ features correspond to $B$ of $(8+i)$, for $1 \leq i \leq 7$. Thus, if we are sure that $B$ = 9, we use the first two dimensions of **Li-14**, i.e. **Li-2**; else when the exact value of $B$ is not known, the 14-dim feature is used.
(iii) **YB-243**: In [11], Yu et al propose the use of a 243-dim feature based on transition probability matrices computed using the difference matrix computed in the pixel and DCT domains.

It is seen that in the lower embedding rate regime for which ME performs better than QIM, these newer features (**KF-548** and **Li-2**) provide similar levels of detectability as that provided by features already discussed, like **PF-274**.

The detection results are also shown for a variety of $QF_h$ in Tables 7 and 8. It is generally seen that the detection accuracy is higher when $QF_h$ = 50 ($QF_a$ is fixed at 75), while it decreases generally as we

Table 7: Comparing $P_{detect}$ for a variety of recently proposed features to detect YASS, using $QF_h$ = 50. We use $B$=9 for the QIM schemes. The acronyms used for the various methods are the same as used in Table 6.

| Feature | QIM-2 | QIM-4 | QIM-6 | QIM-12 | QIM-15 | QIM-19 | (7,3), $B$=9 | (3,2), $B$=9 | (3,2), $B$=10 |
|---|---|---|---|---|---|---|---|---|---|
| KF-548 | 68.45 | 79.48 | 83.82 | 89.20 | 90.44 | 92.03 | 69.61 | 80.15 | 78.97 |
| Li-14 | 54.01 | 55.27 | 56.75 | 59.74 | 62.96 | 67.65 | 52.88 | 59.93 | 55.76 |
| Li-2 | 64.43 | 69.49 | 77.51 | 81.19 | 95.71 | 96.08 | 68.83 | 76.05 | 71.08 |
| YB-243 | 54.64 | 55.70 | 56.75 | 58.12 | 64.01 | 69.89 | 54.23 | 59.68 | 56.12 |

Table 8: Comparing $P_{detect}$ for a variety of recently proposed features to detect YASS, using $QF_h$ = 70. We use $B$=9 for the QIM schemes. The acronyms used for the various methods are the same as used in Table 6.

| Feature | QIM-2 | QIM-4 | QIM-6 | QIM-12 | QIM-15 | QIM-19 | (7,3), $B$=9 | (3,2), $B$=9 | (3,2), $B$=10 |
|---|---|---|---|---|---|---|---|---|---|
| KF-548 | 59.85 | 63.97 | 67.65 | 77.21 | 78.43 | 78.70 | 57.83 | 66.18 | 61.52 |
| Li-14 | 50.49 | 50.67 | 50.85 | 54.17 | 56.00 | 58.70 | 49.94 | 50.00 | 51.35 |
| Li-2 | 53.37 | 58.22 | 71.85 | 76.91 | 79.14 | 81.00 | 58.39 | 69.80 | 53.79 |
| YB-243 | 50.55 | 51.22 | 51.68 | 54.82 | 58.13 | 59.35 | 51.90 | 52.70 | 51.52 |

increase $QF_h$ from 50 to 70.

**Effect of Varying the Size of the Training Dataset:** The training dataset now has 1850 images instead of (1630/2) 815 images, while the test set remains the same. The additional images are generated in the same way as the original set of 1630 images. In Table 9, we observe that there is marginal increase in the detection accuracy after increasing the size of the training dataset by more than a factor of 2.

**Performance Comparison after Puncturing:** We have employed puncturing for the ME-RA framework but not for the QIM-RA scheme. We now use puncturing for QIM-RA ("QIM-RA: $n$ terms" scheme) and compare the bpnc results for ME-RA and QIM-RA, both with and without puncturing, in Table 10. From Table 2 and 3, ME-RA-puncture is less detectable than QIM-RA and also has higher bpnc. After using

Table 9: Comparing $P_{detect}$ for two different sized datasets, using a variety of steganalysis methods, and different variants (embedding methods) of the YASS scheme - $B$=9 is used along with $QF_h$=50 and $QF_a$=75.

| Hiding Scheme | 815 training images | | | | | 1850 training images | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | PF-274 | Chen-486 | KF-548 | Li-2 | YB-243 | PF-274 | Chen-486 | KF-548 | Li-2 | YB-243 |
| QIM-RA: 2 terms | 67.73 | 52.39 | 68.45 | 64.43 | 54.64 | 70.31 | 54.60 | 69.57 | 65.65 | 54.80 |
| QIM-RA: 4 terms | 79.39 | 56.20 | 79.48 | 69.49 | 55.70 | 81.60 | 59.88 | 79.75 | 71.78 | 55.75 |
| QIM-RA: 6 terms | 84.05 | 57.55 | 83.82 | 77.51 | 56.75 | 85.15 | 61.60 | 84.56 | 78.22 | 57.30 |
| ME-RA-puncture (7,3) | 69.45 | 55.95 | 69.61 | 68.83 | 54.23 | 70.35 | 60.86 | 71.66 | 67.47 | 54.40 |
| ME-RA-puncture (3,2) | 78.65 | 61.60 | 80.15 | 76.05 | 59.68 | 78.80 | 62.79 | 81.00 | 78.00 | 60.05 |

Table 10: The bpnc values are compared for **ME-RA** and **QIM-RA** methods, before and after puncturing, at $QF_h$=50.

| Hiding | $B = 9$ | | $B = 10$ | | $B = 25$ | | $B = 49$ | |
|---|---|---|---|---|---|---|---|---|
| Scheme | before | after | before | after | before | after | before | after |
| QIM-RA: 2 terms | 0.0493 | 0.0572 | 0.0382 | 0.0438 | 0.0588 | 0.0670 | 0.0606 | 0.0683 |
| QIM-RA: 4 terms | 0.0864 | 0.0965 | 0.0700 | 0.0784 | 0.1018 | 0.1110 | 0.1048 | 0.1134 |
| QIM-RA: 6 terms | 0.1138 | 0.1206 | 0.0923 | 0.0999 | 0.1336 | 0.1392 | 0.1379 | 0.1427 |
| ME-RA (7,3) | 0.0766 | 0.0975 | 0.0634 | 0.0805 | 0.1000 | 0.1106 | 0.1050 | 0.1136 |
| ME-RA (3,2) | 0.1100 | 0.1200 | 0.0900 | 0.0998 | 0.1300 | 0.1382 | 0.1350 | 0.1421 |

puncturing, we observe that the bpnc gain margin (of ME-RA-puncture over QIM-RA-puncture) decreases - however, in general, ME-RA-puncture is still less detectable (puncturing does not affect the detectability) than QIM-RA-puncture at similar bpnc values. We have also experimentally observed that for "QIM-RA: rand $n$" schemes, the average bpnc is not increased through puncturing.

To conclude, *for hiding conditions where the embedding rate has to be low enough to ensure a certain level of undetectability, ME based embedding with suitable puncturing generally results in higher bpnc than QIM, for similar robustness levels against steganalysis*. However, this holds true only when the channel noise introduced by the active adversary is low enough - for more severe noise, the LLR estimation for ME is erroneous enough to result in a lower hiding rate than QIM.

# References

[1] B. Chen and G. W. Wornell. 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.

[2] C. Chen and Y. Q. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Proc. of International Symposium on Circuits and Systems (ISCAS)*, pages 3029–3032, May 2008.

[3] J. Kodovsky and J. Fridrich. Influence of embedding strategies on security of steganographic methods in the JPEG domain. In *Proc. of SPIE*, pages 2 1 – 2 13, San Jose, CA, Jan. 2008.

[4] J. Kodovský and J. Fridrich. Calibration revisited. In *MM & Sec '09: Proceedings of the 11th ACM workshop on Multimedia and security*, pages 63–74, New York, NY, USA, 2009. ACM.

[5] B. Li, Y. Shi, and J. Huang. Steganalysis of YASS. In *Proceedings of the 10th ACM workshop on Multimedia and security*, pages 139–148. ACM New York, NY, USA, 2008.

[6] T. Pevny and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In *Proc. of SPIE*, pages 3 1 – 3 14, San Jose, CA, 2007.

[7] 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 *Proc. of ICIP*, pages 1292–1295, Oct 2008.

[8] Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In *Lecture notes in computer science: 8th International Workshop on Information Hiding*, pages 249–264, July 2006.

[9] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Robust image-adaptive data hiding based on erasure and error correction. *IEEE Trans. on Image Processing*, 13(12):1627 – 1639, Dec 2004.

[10] K. Solanki, A. Sarkar, and B. S. Manjunath. YASS: Yet Another Steganographic Scheme that resists blind steganalysis. In *9th International Workshop on Information Hiding*, pages 16–31, Jun 2007.

[11] X. Yu and N. Babaguchi. Breaking the YASS Algorithm via Pixel and DCT Coefficients Analysis. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, 2008.