

Perturbed Quantization Steganography

Jessica Fridrich
SUNY Binghamton
Department of ECE
Binghamton, NY 13902-6000
001 607 777 2577
fridrich@binghamton.edu

Miroslav Goljan
SUNY Binghamton
Department of ECE
Binghamton, NY 13902-6000
001 607 777 5793
mgoljan@binghamton.edu

David Soukal
SUNY Binghamton
Department of CS
Binghamton, NY 13902-6000
001 607 777 2577
dsoukal1@binghamton.edu

ABSTRACT

In this paper, we use the recently proposed wet paper codes and introduce a new approach to passive-warden steganography called Perturbed Quantization. In Perturbed Quantization, the sender hides data while processing the cover object with an information-reducing operation that involves quantization, such as lossy compression, downsampling, or A/D conversion. The unquantized values of the processed cover object are considered as side information to confine the embedding changes to those unquantized elements whose values are close to the middle of quantization intervals. This choice of the selection channel calls for wet paper codes as they enable communication with non-shared selection channel. Heuristic is presented that indicates that the proposed method provides better steganographic security than current JPEG steganographic methods. This claim is further supported by blind steganalysis of a specific case of Perturbed Quantization for recompressed JPEG images.

Keywords

Perturbed quantization, wet paper code, adaptive, security, steganography

1. MOTIVATION

The primary goal of steganography is to build a statistically undetectable communication channel (the famous Prisoner Problem [1]). In order to embed a secret message, the sender slightly modifies the cover object to obtain the embedded stego object. In steganography under the passive warden scenario [2,3], the goal is to communicate as many bits as possible without introducing any detectable artifacts into the cover object. Attempts to give a formal definition of the concept of steganographic security can be found in [4–6]. In practice, a steganographic scheme is considered secure if no existing attack can be modified to build a detector that would be able to distinguish between cover and stego images with a success better than random guessing [7].

One possible measure to improve the security of steganographic schemes for digital media is to embed the message in adaptively selected components of the cover object [8–10], such as noisy areas or segments with a complex texture. However, if the adaptive selection rule is public or only “weakly dependent on a key”, the attacker can apply the same rule and start building an attack. It is then a valid question whether the adaptive selection improves steganographic security at all. An interesting example of

a scheme where adaptive pixel selection in fact decreased its security is the recent surprising result of Westfeld [11].

This problem with adaptive steganography could be remedied if the selection rule was determined from some side information available only to the sender but *in principle unavailable* to the attacker. For example, imagine the situation when the sender has a raw, uncompressed image and wants to embed data into its JPEG compressed form. Can the sender use his side information – the uncompressed image – to construct a better JPEG steganography? We can attempt to select for embedding those DCT coefficients whose unquantized values lie “close to the middle” of quantization intervals. Intuitively, perturbing the rounding process at such coefficients will be harder to detect than modifying the coefficients that experienced a small rounding error during quantization. The obvious problem of this proposition, however, is that the recipient will not know from which coefficients to read the message.

There are other situations in steganography that call for a general solution to the problem of communication with non-shared or partially shared selection channels. The so called “wet paper codes” were recently proposed as a general method that enables steganography with a non-shared selection rule (a.k.a. writing on wet paper) [12,13]. Because wet paper codes enable the sender to communicate to the recipient on average *the same number of bits as if the receiver knew the set of dry pixels*, the above mentioned problem of adaptive steganography is removed.

In Section 2, we describe a new general approach to steganography called Perturbed Quantization and propose several practical embedding scenarios. To make this paper self-contained, in Section 3 we briefly describe wet paper codes based on random linear codes and their practical implementation. In Section 4, we introduce Perturbed Quantization steganographic method for JPEG images that embeds message bits while recompressing a JPEG image with a lower quality factor. Security of this new technique is analyzed in Section 5, where we report the results of blind steganalyzers and compare the results with current state of the art JPEG steganographic algorithms. Summary and future directions are given in Section 6.

2. PERTURBED QUANTIZATION

We explain the basic idea on the example mentioned in the introduction. Let us assume that the sender has a raw grayscale image X that has never been compressed before. During JPEG compression, the Discrete Cosine Transform (DCT) is performed, the DCT coefficients are divided by quantization steps from the

quantization table, then rounded to integers, and finally encoded according to the JPEG standard to a JPEG file, G . Let us denote the DCT coefficients (divided by quantization steps) before and after rounding with d_i and D_i , respectively, $i = 1, \dots, n$, where n is the total number of DCT coefficients in G . Identify those coefficients d_i whose fractional part is in a narrow interval around 0.5, $d_i - \lfloor d_i \rfloor \in [0.5 - \varepsilon, 0.5 + \varepsilon]$, where ε is called tolerance and should be set to a small number (e.g., $\varepsilon = 0.1$ or smaller). Such coefficients will be called *changeable coefficients*. The symbol $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x .

Let $S = \{i_1, \dots, i_k\}$ be the set of indices of all changeable coefficients. During compression, we will round changeable coefficients $d_j, j \in S$, up or down at our will and thus encode up to $k = |S|$ bits (obtaining a compressed and embedded image G'). However, we cannot simply code the message bits as parities (for example LSBs) of the rounded DCT coefficients D_j because the recipient would not know which coefficients carry message bits. Instead, we use the wet paper codes that give us the opportunity to communicate via non-shared selection channels (see Section 3).

We call this method Perturbed Quantization (PQ) because during compression we slightly perturb the quantizer (the process of rounding to integers) for a certain subset of changeable coefficients in order to embed message bits. It is easy to show that the difference between the average rounding distortion of the regular quantizer and its perturbed form is ε^2 , which is at least by an order of magnitude smaller than the average rounding error (1/4). An attacker would have to be able to find statistical evidence that some of the values D_i were quantized “incorrectly”. This is likely going to be a formidable task for the following heuristic reasons:

- (1) The sender is using side information that is largely removed during quantization and is thus unavailable to the attacker. It is in general impossible to reverse JPEG compression and obtain the uncompressed image or a good approximation to the uncompressed image. On the other hand, in some areas of the image (e.g., with a smooth gradient), one might be able to obtain a better approximation and attempt to attack PQ.
- (2) Thus, the sender can (and should) accept additional selection rules to exclude from the set S those coefficients whose unquantized values can be predicted with better accuracy.
- (3) The actual rounding of values d_i is more influenced by the image noise for changeable coefficients than for the remaining coefficients because the changeable coefficients are close to the middle of the rounding intervals. As a result, the rounding process $d_i \rightarrow D_i$ has a large stochastic component.

2.1 Information-reducing processes

The idea outlined above can be formulated in a more general setting. Whenever the sender downgrades a digital image using lossy compression, downsizing, quantization, format conversion, recompression, etc., he will have access to all numerical values before quantization/rounding occurs. Thus, the sender gains the same ability to slightly modify the rounding process whenever he subjects the cover image to an *information-reducing* process that involves a real transform followed by a quantizer/rounding. The heuristic is that because the process is information-reducing, an attacker cannot easily recover from the stego image those fine

details of the original image that would enable him to mount a reliable attack.

In the rest of this paper, boldface symbols will be used to denote matrices and vectors. Let us assume that the cover image \mathbf{x} is represented with a vector $\mathbf{x} \in I^m$, where I is the range of its m elements (pixels, coefficients, colors, indices) depending on the format of \mathbf{x} . For example, for an 8-bit grayscale image, $I = \{0, \dots, 255\}$. The information-reducing process F is modeled as a transformation

$$F = Q \circ T: I^m \rightarrow J^n, \quad (1)$$

where J is the integer dynamic range of the downgraded image $\mathbf{y} = F(\mathbf{x})$ represented with an n -dimensional integer vector $\mathbf{y} \in J^n$, $m \geq n$. The transform $T: I^m \rightarrow \mathbf{R}^n$ is a real-valued transformation and $Q: \mathbf{R}^n \rightarrow J^n$ is a quantizer. The intermediate “image” $T(\mathbf{x})$ will be represented using an n -dimensional vector $\mathbf{u} \in \mathbf{R}^n$. We give several examples of image downgrading operations F that could be used for PQ steganography.

Example 1 (Resizing). For grayscale images, the transformation T maps a square $m_1 \times m_2$ matrix of integers $(x_{ij}), i=0, \dots, m_1-1, j=0, \dots, m_2-1$ into an $n_1 \times n_2$ matrix of real numbers $(u_{rs}), r=0, \dots, n_1-1, s=0, \dots, n_2-1, m_1 > n_1, m_2 > n_2$, using a resampling algorithm. The quantizer Q is a uniform integer quantizer (rounding to integers) applied to the vector (matrix) \mathbf{u} by coordinates

$$Q(u_{rs}) = \text{round}(u_{rs}). \quad (2)$$

Example 2 (Decreasing the color depth by d bits). The transformation T maps a square $m_1 \times m_2$ matrix of integers (x_{ij}) in the range $I = \{0, \dots, 2^b - 1\}, i=0, \dots, m_1-1, j=0, \dots, m_2-1$ into an $m_1 \times m_2$ matrix of real numbers $(u_{ij}), u_{ij} = x_{ij} / 2^d$. The quantizer Q is the same uniform scalar quantizer as in Example 1.

Example 3 (JPEG compression). For grayscale images, the transformation T maps a square $m_1 \times m_2$ matrix of integers (x_{ij}) , into a $\lceil m_1/8 \rceil \times \lceil m_2/8 \rceil$ matrix of real numbers (u_{ij}) in a block-by-block manner ($\lceil x \rceil$ denotes the smallest integer larger than or equal to x). In each 8×8 pixel block \mathbf{B} , the corresponding block in u_{ij} is $\text{DCT}(\mathbf{B})/q$, where DCT is the 2D DCT transform, q is the quantization matrix, and the operation “./” is an element-wise division. The quantizer Q is, again, given by (2).

Continuing the description of Perturbed Quantization, the sender identifies the set of indices $S \subset \{1, \dots, n\}$ of object elements whose values $u_j, j \in S$, may be perturbed during quantization. The set S will be determined using some Selection Rule (SR). There are no restrictions on the form of the rule. The sender can use his knowledge of \mathbf{x} and \mathbf{u} , which are unavailable to the receiver or any attacker. As already mentioned above, the sender can, for example, select u_i whose values are close to the middle of the quantization intervals of Q

$$S = \{i \mid i \in \{1, \dots, n\}, u_i \in [L + 0.5 - \varepsilon, L + 0.5 + \varepsilon] \text{ for some integer } L\}. \quad (3)$$

The tolerance ε could in principle be adaptive and depend on the neighborhood of the element x_i . It can also be made key

dependent if desired. In this paper, we assume for simplicity that ε is a publicly known small constant. The sender will communicate a message to the receiver by rounding changeable elements u_j , $j \in S$, to either L or $L+1$ and rounding all other elements u_i , $i \notin S$, using the quantizer (2), $y_i = Q(u_i)$.

We note that the selection rule does not have to necessarily be of the type (3) and can be defined differently based on other heuristic depending on the format of \mathbf{x} and properties of its elements. In Section 4, we give an example of a slightly different SR for the situation when the information-reducing transformation is recompression of the cover JPEG image using a lower quality factor.

Once the changeable elements have been identified, the sender needs to encode the message bits. Let \mathbf{p} , $p_i = P(y_i)$, be the vector of element parities¹ for the processed cover object $\mathbf{y} = F(\mathbf{x})$. By perturbing the rounding process as described above, the sender can modify k bits p_j , $j \in S$, but cannot modify the remaining $n - k$ bits. The recipient does not know the set S . This is an example of a channel known as an n -bit memory with up to $n - k$ defective cells introduced in 1974 by Kuznetsov et al. [14]. It is known that the Shannon capacity of this channel is k/n [15–17] and can be achieved for non-binary alphabets using an algebraic coding scheme with the cosets of an MDS code as bins [16]. The same paper contains a noisy generalization of this channel and shows that nested linear codes (or “partitioned” codes) are capable of achieving the theoretical maximum capacity.

In steganographic applications, the number of defective cells may be quite large. For example, in the double compression embedding described in Section 4, for a typical JPEG image, $n \sim 10^6$ and $k \sim 10^4$. Furthermore, the number of stuck cells can vary greatly among different covers and across embedding schemes, which makes application of fixed rate codes more complicated. Reflecting on these specifics of steganographic applications, so called wet paper codes were proposed in the past as an efficient coding approach to this channel [12,13]. For completeness, we briefly describe the basic ideas behind wet paper codes and their implementation in the next section.

3. Wet paper code

To explain this metaphor, imagine a situation when the cover object (a digital image, for example) has been exposed to “rain” and the sender can only slightly modify the dry spots of the cover image but not the wet spots. During transmission, the stego image dries out and thus the recipient does not know which pixels the sender used. We note that in this scenario we allow the rain to be truly random, pseudo-random, completely determined by the sender or the image, or an arbitrary mixture of all of the above. This channel is obviously equivalent to writing in memory with defective cells by identifying wet pixels with defective cells.

¹ The parity could be any function defined on J with range $\{0,1\}$ such that $P(k) = 1 - P(k+1)$ for all $k \in J$. Thus, for J consisting of consecutive integers, only two parity functions are possible, $P_1(k) = \text{LSB}(k)$ or $P_2(k) = 1 - \text{LSB}(k)$ (the shifted LSB). The Parity function could be the same for all elements or chosen randomly between P_1 or P_2 for each element based on a secret key.

3.1 Encoder and decoder

The wet paper code can be viewed as a generalization of the selection channel [3] where one message bit is embedded as the parity of a group of cover object elements. In the selection channel, at most one element value must be changed in order to match the parity of a group of elements to the message bit. The parity of the group is a sum modulo 2 of the individual element parities. If there are m elements that can be changed in the group, one can attempt to embed m message bits by forming m linearly independent linear combinations of element parities instead of just one sum.

Let us assume that the sender wants to communicate M bits $\mathbf{b} = \{b_1, \dots, b_M\}^T$. At this point, we assume that the recipient knows M . Later, we show how to modify the communication scenario to the case when the recipient does not know M . The sender and recipient agree on a secret stego key that is used to generate a pseudo-random binary matrix \mathbf{H} of dimensions $M \times n$. The sender will round u_j , $j \in S$, obtaining the column vector \mathbf{y}' , so that the modified parity column vector $\mathbf{p}' = P(\mathbf{y}')$ satisfies

$$\mathbf{H}\mathbf{p}' = \mathbf{b}. \quad (4)$$

Thus, the sender needs to solve a system of linear equations in Galois Field GF(2). This setup is an example of coset coding. The message is communicated as a syndrome with parity check matrix \mathbf{H} . As opposed to the approach by Heegard [15] who proposed a fixed rate code for memory with defective cells, we use a variable rate random linear code with a pseudo-random matrix. This randomization offers flexibility for our steganographic application in which k varies greatly depending on the cover object and a steganographic method. Finally, we note that the selection channel [3] is a special case of (4) when $\mathbf{H} = [1 \ 1 \ \dots \ 1]$ is a single-row $1 \times n$ matrix.

The sender sends the modified stego object \mathbf{y}' to the recipient. The decoding is very simple because the recipient first forms the vector $\mathbf{p}'_i = P(y'_i)$ and then evaluates $\mathbf{H}\mathbf{p}'$ using the shared matrix \mathbf{H} . The extracted message is simply $\mathbf{b} = \mathbf{H}\mathbf{p}'$.

3.2 Average capacity

It will be advantageous to rewrite (4) to

$$\mathbf{H}\mathbf{v} = \mathbf{b} - \mathbf{H}\mathbf{p} \quad (5)$$

using the variable $\mathbf{v} = \mathbf{p}' - \mathbf{p}$. In the system (5), there are k unknowns v_j , $j \in S$, while the remaining $n - k$ values v_i , $i \notin S$, are zeros. Thus, on the left hand side, we can remove from \mathbf{H} all $n - k$ columns i , $i \notin S$, and also remove from \mathbf{v} all $n - k$ elements v_i with $i \notin S$. Keeping the same symbol for \mathbf{v} , (5) now becomes

$$\bar{\mathbf{H}}\mathbf{v} = \mathbf{b} - \mathbf{H}\mathbf{p}, \quad (6)$$

where $\bar{\mathbf{H}}$ is a binary $M \times k$ submatrix of \mathbf{H} and \mathbf{v} is an unknown $k \times 1$ binary vector. This system has a solution for an arbitrary

message \mathbf{b} as long as $\text{rank}(\bar{\mathbf{H}})=M$. The probability that a random² $M \times k$ binary matrix has rank M is $1-O(2^{M-k})$ [18, follows from Lemma 4], which quickly approaches 1 for fixed k as M decreases to zero. This suggests that the expected maximal number of bits M_{\max} that can be communicated is likely close to k . In fact, assuming that the sender keeps adding rows to \mathbf{H} while (6) still has a solution, the maximal number of bits that the sender can communicate is [12] (assuming $n \geq k, k \rightarrow \infty$)

$$M_{\max}(k) = k + O(2^{-k/4}). \quad (7)$$

This means that on average, the sender will be able to communicate k bits to the recipient using the wet paper code.

We now explain how to relax the assumption that the recipient knows k or M . The sender and recipient can generate the matrix \mathbf{H} in a row-by-row manner rather than generating it as a two-dimensional array of $M \times n$ bits. In this way, the sender can reserve the first few bits of the message \mathbf{b} for a header of length $\lceil \log_2(n) \rceil$ bits to inform the recipient of the number of rows in \mathbf{H} . The recipient first generates the first $\lceil \log_2(n) \rceil$ rows of \mathbf{H} , multiplies them by the received vector \mathbf{p}' , and reads the header (the message length M). Then, he generates the rest of \mathbf{H} , and reads the message $\mathbf{b} = \mathbf{H}\mathbf{p}'$. Thus, under the assumption that the recipient has no information about either k or M , the sender can on average communicate $k - \log_2 n$ bits.

3.3 Practical implementation

As the focus of this paper is a specific application of wet paper codes – Perturbed Quantization steganography – we only briefly summarize approaches published elsewhere [12,13].

The main complexity of the wet paper code is on the sender's side, who needs to solve M linear equations (6) for k unknowns in GF(2). Assuming that the maximal length message $M = k$ is sent, the complexity of Gaussian elimination is $O(k^3)$, which would lead to impractical performance for large payloads, such as $k > 10^5$. To overcome this cubic complexity, we can divide the cover object into n/n_B disjoint random subsets (determined from the shared stego key) of a fixed, predetermined size n_B and then perform the embedding for each subset separately. The complexity of embedding is now proportional to $n/n_B(kn_B/n)^3 = nr^3 n_B^2$, where $r = k/n$ is the rate, and is thus linear in the number of cover object elements, albeit with a large constant.

A different possibility to realize the wet paper code is to impose a special stochastic structure on the columns of \mathbf{H} and use the LT process [19] to solve (6) in a much more efficient manner. In particular, if the Hamming weights of columns of \mathbf{H} follow so called Robust Soliton Distribution (RSD), it can be shown that $\bar{\mathbf{H}}$ can be brought into an upper diagonal form $[\mathbf{U}, \mathbf{H}']$ by permuting its columns and rows (the matrix LT process). Here, \mathbf{U} is a square $M \times M$ upper triangular matrix with ones on its main diagonal and \mathbf{H}' is a binary $M \times (k-M)$ matrix. The details of this approach are given in [13], where it is shown that the process of permuting the rows and columns of $\bar{\mathbf{H}}$ is equivalent to the LT process on a bipartite graph with bi-adjacency matrix $\bar{\mathbf{H}}$. Once

$\bar{\mathbf{H}}$ is in this form, solving the system $[\mathbf{U}, \mathbf{H}']\mathbf{v} = \mathbf{b}$ is quite easy as the solution is simply obtained using regular back substitution as in Gaussian elimination.

The RSD is defined below. The probability that the Hamming weight of a column in \mathbf{H} is $i, 1 \leq i \leq M$, is $(\rho_i + \tau_i)/\beta$, where

$$\rho_i = \begin{cases} \frac{1}{M} & i = 1 \\ \frac{1}{i(i-1)} & i = 2, \dots, M \end{cases},$$

$$\tau_i = \begin{cases} R/(iM) & i = 1, \dots, M/R - 1 \\ R \ln(R/\delta)/M & i = M/R \\ 0 & i = M/R + 1, \dots, M \end{cases}, \quad (8)$$

$$\beta = \sum_{i=1}^M (\rho_i + \tau_i), \text{ and } R = c \ln(M/\delta) \sqrt{M}$$

for some suitably chosen constants δ and c . The RSD was designed [19] so that the probability that $\bar{\mathbf{H}}$ can be brought into the upper diagonal form by permuting its rows and columns is better than $1-\delta$ as long as the number of changeable elements k satisfies

$$k > \beta M = M + O(\sqrt{M} \ln^2(M/\delta)). \quad (9)$$

This means that there is a small capacity loss of $O(\sqrt{M} \ln^2(M/\delta))$ in exchange for solving (6) quickly using the matrix LT process. The capacity loss, together with probability of successful pass through the matrix LT process for different values of k are shown in Table 1. We see that the probability of successful pass increases while the capacity loss decreases with increasing k (the larger the problem, the better this method works).

k	Gauss	LT	β	P
1000	0.023	0.008	1.098	43%
10000	17.4	0.177	1.062	75%
30000	302	0.705	1.047	82%
100000	9320	3.10	1.033	90%

Table 1 Running time (in seconds) for solving $k \times k$ and $k \times \beta k$ linear systems using Gaussian elimination and matrix LT process ($c = 0.1, \delta = 5$); P is the probability of a successful pass.

Table 1 also shows the performance comparison between solving (6) using Gaussian elimination and the matrix LT process. The LT process enables solving the system as a whole at once, which greatly simplifies implementation and decreases computational complexity at the same time.

Before changing the subject to PQ, we briefly address two more issues. First, the RSD depends on the message length M and thus it needs to be somehow communicated to the recipient. This can be arranged, for example, by dividing the cover object into two

² The probability of 0 and 1 in \mathbf{H} is the same and equal to 1/2.

disjoint parts. The first one is large enough³ just to communicate the message length (e.g., up to 20 bits) using a pseudo-random matrix \mathbf{H}_0 (generated from the stego key) whose elements are iid realizations of a uniform binary variable. This system is solved using Gaussian elimination, which is fast as \mathbf{H}_0 only has few rows. The message embedding in the second subset is done using the matrix LT process with now known value of M . The parameters c and δ are either publicly known fixed constants or may depend on M .

The second issue is what to do when the matrix LT process fails to bring $\bar{\mathbf{H}}$ into an upper diagonal form. This can be solved by generating the matrix \mathbf{H} from a seed that is the stego key concatenated with a few bits (e.g., say w bits) that are communicated together with the message length M . The sender simply tries to solve the system with those additional w bits all set to 0 and if the LT process does not pass, the sender changes the bits (there are 2^w possibilities) till a successful pass is obtained. Given that the probability of a successful pass is, say, 0.8, one can see that in practice, $w \sim 5$ bits should suffice).

4. EMBEDDING WHILE DOUBLE COMPRESSING

In this section, we apply Perturbed Quantization to the information-reducing process of repeated JPEG compression. First, we introduce the necessary basics of JPEG compression, then explain the embedding method and calculate its capacity. In Section 5, we subject this method to blind steganalysis and compare its performance to existing methods. We further note that due to simplicity we work with grayscale images. The considerations hold for color images as well.

4.1 JPEG compression preliminaries

In JPEG compression, the image is first divided into disjoint blocks of 8×8 pixels. For each block \mathbf{B}^x (with integer pixel values in the range 0–255), the discrete cosine transform, $c = DCT(\mathbf{B}^x)$, produces 64 DCT coefficients (c_{ij}), $0 \leq i, j \leq 7$, which are then divided using the quantization matrix $\mathbf{q}=(q_{ij})$ and rounded to integers using the quantizer (2)

$$\begin{aligned} c_{ij} &= \sum_{k,l=0}^7 a_{kl}(i,j) B_{kl}^x \\ d_{ij} &= c_{ij} / q_{ij} \\ D_{ij} &= Q(d_{ij}). \end{aligned} \quad (10)$$

In (10), $a_{kl}(i,j)$ are the elements of the DCT transform matrix

$$\begin{aligned} a_{kl}(i,j) &= \frac{1}{4} w(k)w(l) \cos \frac{\pi}{16} k(2i+1) \cos \frac{\pi}{16} l(2j+1), \quad (11) \\ w(k) &= 1/\sqrt{2} \text{ when } k=0 \text{ and } w(k)=1 \text{ otherwise.} \end{aligned}$$

The quantized coefficients D_{ij} are arranged in a zigzag manner and compressed using the Huffman encoder. The resulting compressed stream together with a header forms the final JPEG file.

The JPEG decompression works in the opposite order. The JPEG bit-stream is decompressed using the Huffman decoder and, for each block, each quantized DCT coefficient D_{ij} is multiplied by q_{ij} , the whole block is then inverse DCT transformed, and the result is rounded and clipped to a finite dynamic range obtaining the 8×8 pixel block \mathbf{B} in the decompressed image

$$\begin{aligned} C_{ij} &= q_{ij} D_{ij} \\ \mathbf{B}^{raw} &= DCT^{-1}(C) \\ \mathbf{B} &= [\mathbf{B}^{raw}], \end{aligned} \quad (12)$$

where $[x] = Q(x)$ for $0 \leq x \leq 255$, $[x] = 0$ for $x < 0$, and $[x] = 255$ for $x > 255$.

Let us assume that the cover JPEG file has been decompressed to the spatial domain to image \mathbf{x} . Let \mathbf{B} be an 8×8 block in \mathbf{x} . Assuming that \mathbf{B} has no pixels saturated at 0 or 255, from (12) we see that the quantization error $\xi_{ij} = B_{ij}^{raw} - B_{ij}$, $0 \leq i, j \leq 7$, satisfies $-0.5 \leq \xi_{ij} \leq 0.5$. Consequently,

$$DCT(\mathbf{B}) = DCT(\mathbf{B}^{raw}) - DCT(\boldsymbol{\xi}) = \mathbf{C} - \boldsymbol{\eta}, \quad (13)$$

where $\eta_{ij} = \sum_{k,l=0}^7 a_{kl}(i,j) \xi_{kl}$.

Modeling the quantization error ξ_{ij} as an i.i.d. noise uniform on $(-1/2, 1/2]$, we obtain

$$\begin{aligned} E(\eta_{ij}) &= \sum_{k,l=0}^7 a_{kl}(i,j) E(\xi_{kl}) = 0, \\ E(\eta_{ij}^2) &= \sum_{k,l=0}^7 a_{kl}^2(i,j) E(\xi_{kl}^2) + \\ &\sum_{k,l=0}^7 \sum_{\substack{r,s=0 \\ (r,s) \neq (k,l)}}^7 a_{kl} a_{rs} E(\xi_{kl} \xi_{rs}) = \frac{1}{12} \end{aligned}$$

because $E(\xi_{ij}^2) = 1/12$ and $\sum_{k,l=0}^7 a_{kl}^2(i,j) = 1$ for all i, j due to the fact that the DCT is an orthonormal transformation. Finally, because η_{ij} is an average of bounded independent variables, by the Liapunov extension of the Central Limit Theorem (see, for example [20]), the distribution of η_{ij} is approximately Gaussian $N(0, 1/12)$.

³ Its size is determined by the smallest rate k/n one can encounter for a given stego scheme.

4.2 Effects of repeated JPEG compression and the embedding algorithm

In this section, we investigate the impact of double compression on distribution of DCT coefficients and explain how double compression can be used in the context of Perturbed Quantization. Let us assume that we have an image that is a decompressed JPEG with quality factor Q_1 (with quantization matrix $q_{ij}^{(1)}$) and we resave it as JPEG again but with a different quality factor Q_2 (with quantization matrix $q_{ij}^{(2)}$). For simplicity, we take a look at a specific DCT coefficient with $(i, j) = (1, 2)$ (the first AC coefficient in the zigzag scan) and $Q_1 = 88$, $Q_2 = 76$. In the original JPEG image, the DCT values C_{12} are multiples of $q_{12}^{(1)} = 3$ (see the top part of Figure 1. As explained above, after decompression (12) and the second DCT transform (10), the values of c_{12} will no longer be exact multiples of 3 but will be spread around them as in the bottom part of Figure 1. Next, we look at what happens when the coefficients c_{12} are quantized with a quantization step $q_{12}^{(2)} = 6$ corresponding to the second quality factor $Q_2 = 76$.

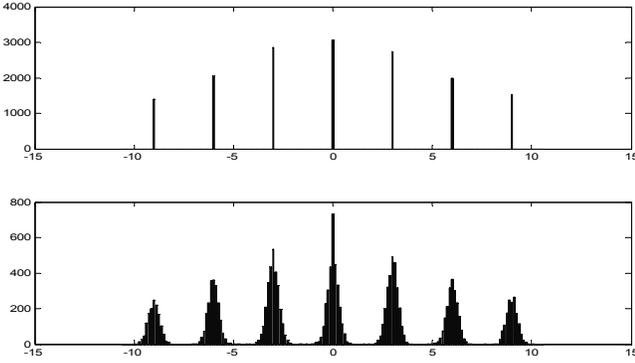


Figure 1 Top: histogram of values of the DCT coefficient C_{12} in the original 88% quality JPEG file (note that the values are multiples of the quantization step $q_{12}^{(1)} = 3$). **Bottom:** histogram of the same DCT coefficient c_{12} after decompressing the JPEG file to the spatial domain and DCT transforming.

From Figure 1, one can see that the peaks around the even multiples $2k \times 3$, $k = \dots, -1, 0, 1, \dots$, are quantized to $6k$, while the peaks around the odd multiples $(2k+1) \times 3$, $k = 0, 1, \dots$, are split in half, the “left” half being quantized to $6k+2$ and the right half to $6k+4$. Based on the arguments presented in the previous section, this quantization during a normal double compression is essentially a random process because η_{12} is Gaussian $N(0, 1/12)$. This gives us the possibility to build a Perturbed Quantization embedding method by including *all* odd multiples $(2k+1) \times 3$ to the set of changeable coefficients. In the next section, we formulate the Selection Rule for an arbitrary combination of quantization matrices $q^{(1)}$ and $q^{(2)}$.

4.3 Coefficient selection rule

We can use other DCT coefficients c_{ij} for embedding as long as the first and the second quantization steps $q_{ij}^{(1)}$ and $q_{ij}^{(2)}$ satisfy certain numerical properties. The pair $(q_{ij}^{(1)}, q_{ij}^{(2)})$ will be called contributing if there exist integers k and l such that

$$kq_{ij}^{(1)} = lq_{ij}^{(2)} + q_{ij}^{(2)}/2. \quad (14)$$

All integers k and l , $l+1$ that satisfy (14) will be called contributing multiples of $q_{ij}^{(1)}$ and $q_{ij}^{(2)}$, respectively. The condition says that the pair $(q_{ij}^{(1)}, q_{ij}^{(2)})$ is contributing if there exists a multiple of $q_{ij}^{(1)}$ (a contributing multiple) that is exactly in the middle of the second quantization interval of length $q_{ij}^{(2)}$. The following theorem gives a sufficient and necessary condition for the pair $(q_{ij}^{(1)}, q_{ij}^{(2)})$ to be contributing and also gives a formula for all contributing multiples of $q_{ij}^{(1)}$.

Theorem 1. *The pair $(q_{ij}^{(1)}, q_{ij}^{(2)})$ is contributing if and only if $q_{ij}^{(2)}/g$ is even, where $g = \text{GCD}(q_{ij}^{(1)}, q_{ij}^{(2)})$ is the greatest common divisor of $q_{ij}^{(1)}$ and $q_{ij}^{(2)}$. Furthermore, all contributing multiples k of $q_{ij}^{(1)}$ are expressed by the formula*

$$k = (2m+1) \frac{q_{ij}^{(2)}}{2g}, \quad m = \dots, -2, -1, 0, 1, 2, \dots \quad (15)$$

Proof. The implication from left to right is trivial. Dividing (14) by g gives $q_{ij}^{(2)}/2g = kq_{ij}^{(1)}/g - lq_{ij}^{(2)}/g$. Because there is an integer on the right hand side, $q_{ij}^{(2)}/(2g)$ is an integer, too. To prove the other implication, from the Euclid theorem [21], there are two integers a and b such that $aq_{ij}^{(1)} + bq_{ij}^{(2)} = g$. After multiplying this equation by $q_{ij}^{(2)}/(2g)$, which is an integer, we obtain (14) with $k = aq_{ij}^{(2)}/(2g)$ and $l = -bq_{ij}^{(2)}/(2g)$. To derive the formula (15), from (14) we have

$$k = \frac{(2l+1)q_{ij}^{(2)}}{2q_{ij}^{(1)}} = \frac{(2l+1) \frac{q_{ij}^{(2)}}{2g}}{\frac{q_{ij}^{(1)}}{g}}. \quad (16)$$

Because $\text{GCD}(q_{ij}^{(1)}/g, q_{ij}^{(2)}/g) = 1$, it must be the case that $2l+1$ is an odd multiple of $q_{ij}^{(1)}/g$ (note that $q_{ij}^{(1)}/g$ must be odd). Thus, the contributing multiples of $q_{ij}^{(1)}$ are odd multiples of $q_{ij}^{(2)}/(2g)$. This ends the proof. \square

All contributing coefficients in the single compressed JPEG cover image form the set of changeable coefficients S . Theorem 1 can be used to calculate the cardinality of S . Let $h_{ij}(k)$ be the histogram of the DCT coefficient C_{ij} of the cover JPEG file (the one compressed with $q_{ij}^{(1)}$). The number of changeable coefficients $|S|$ is given by the following formula

$$|S| = \sum_{i,j=0}^7 \sum_k z_{ij} h_{ij} \left((2k+1) \frac{q_{ij}^{(2)}}{2g} \right), \quad (17)$$

where $z_{ij} = 1$ if $(q_{ij}^{(1)}, q_{ij}^{(2)})$ is a contributing pair and $z_{ij} = 0$ otherwise.

To show how $|S|$ depends on the quality factors Q_1 and Q_2 , we evaluated (17) for all combinations of quality factors ranging from 50 to 95. The result was averaged over 20 test grayscale images and is displayed in Figure 3. The plot shows that one can choose from a variety of combinations of both quality factors to

achieve a relatively large capacity up to 0.5 bits per non-zero DCT coefficient of the stego image (bpc). Note the ridge of high capacities corresponding to $Q_2 = 2(Q_1 - 50)$. This combination of quality factors translates to $q_{ij}^{(2)} = 2q_{ij}^{(1)}$ (as in Figure 2).

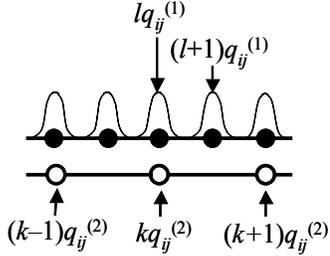


Figure 2 Example of a contributing multiple.

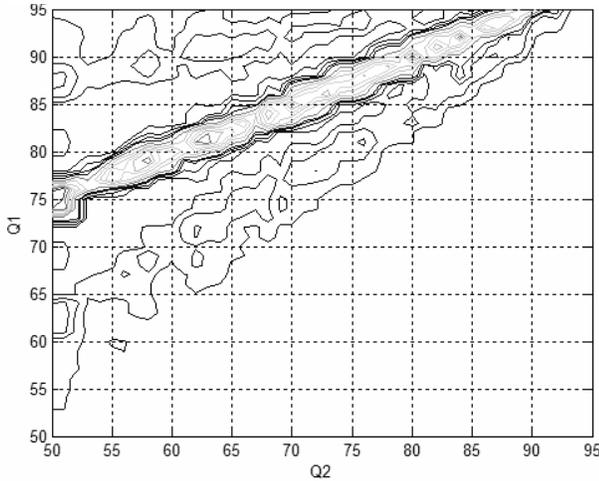


Figure 3 Embedding capacity expressed in bpc (bits per non-zero DCT coefficient of the double-compressed image) averaged over 20 test images. Note the prominent ridge with peaks at $\text{bpc} \approx 0.4$ for quality factors satisfying $Q_2 = 2(Q_1 - 50)$.

4.4 Encoder summary

We summarize the PQ embedding method based on double compression. The method takes a (single compressed) JPEG file as the cover image and produces a double compressed and embedded JPEG file as the stego image. The sender and recipient can use the LSB of DCT coefficients as the parity function. The sender chooses the second quality factor $Q_2 < Q_1$ (to make the recompression information-reducing) so that the number of secret message bits is within the capacity (17) with some reserve for the headers and identifies the set S of changeable coefficients c_{ij} from the quantization matrices $q^{(1)}$ and $q^{(2)}$ using Theorem 1. From (14), the sender enforces that after the second JPEG compression, the quantized value D_{ij} (10) of the ij -th *changeable* DCT coefficient in the stego file is either l or $l+1$, where $kq_{ij}^{(1)} = lq_{ij}^{(2)} + q_{ij}^{(2)}/2$ and k is the value of the quantized ij -th DCT coefficient in the cover image. The sender remembers the values l and $l+1$ for each changeable coefficient c_{ij} and uses them as two possible values for D_{ij} in the stego JPEG file. The embedding process continues with decompression of the cover JPEG file to

the spatial domain and recompression with the second quantization table. This determines the values of all coefficients that are *not changeable*. The value D_{ij} of each changeable coefficient is determined during the embedding process using wet paper codes.

To cast the embedding in the setup of Section 2.1, the transform $F = Q \circ T$ is composed of the decompression (12), the DCT transform (10), division by the second quantization matrix $q^{(2)}$, and the quantizer Q (2). Symbolically, for each 8×8 block D of quantized DCT coefficients from the cover image,

$$T(D) = DCT([DCT^{-1}(q^{(1)} \cdot D)] ./ q^{(2)}), \quad (18)$$

where $q^{(1)} \cdot D$ is the element-wise product of both matrices, “./” is the element-wise division, and $DCT^{-1}(B)$ is the inverse DCT of the coefficient block B .

5. STEGANALYSIS

In this section, we investigate the character of the embedding distortion and evaluate the security of the proposed algorithm using blind steganalyzers.

First of all, we would like to point out that double compressed images are not that unusual, as it might seem at the first sight. Vast majority of owners of digital cameras use the JPEG format for storing images inside the camera. Then, as the images are downloaded to the computer, they may be processed and resaved as JPEGs in some image processing software with a default or a user-specified quality factor. Because most digital cameras adjust the quantization table to the image (to guarantee that all images have approximately the same size), digital camera images have a wide range of quality factors and quantization tables. There are several cases when the user will frequently (unconsciously) create a double-compressed image that will be double-compressed in a manner compatible with our steganographic scheme: The user

1. rotates it by 90 degrees and resaves (it is easy to see that for images whose dimensions are multiples of 8, during rotation by multiples of 90 degrees, each DCT coefficient D_{ij} may either not change or change to D_{ji} and/or change its sign), or
2. recompresses the image with a lower quality factor to decrease its size (e.g., for sending by e-mail) or
3. removes the red eye glare (a few dozen pixels) and resaves the image as JPEG, or
4. adjusts the brightness and resaves.

Thus, we believe that double-compressed images are, in fact, quite ubiquitous and should not be suspicious by themselves. We stress that if the image is resized or cropped by non-multiples of 8 before resaving, or modified in any way that removes the quantized structure of DCT coefficients, we do not call the image a double compressed image because it will not exhibit traces of repetitive compression in the sense of this paper. In this case, one may use the approach from Example 3 from Section 2 for embedding.

We point out that *it is necessary that the second quality factor be smaller than the first one, $Q_1 > Q_2$* . If the second quality factor was larger than the first one, one could first estimate the first quantization table using methods in [22] and then exactly recover the single compressed cover image (compressed with Q_1). In fact, this property of double JPEG compression is used in some semi-fragile watermarking systems for content authentication [23]. Once this single compressed image is obtained, the attacker will simply recompress it with Q_2 and compare to the stego image. Any discrepancies will be indicative of steganography. This attack can be mounted because the double compression is *not* information-reducing when $Q_1 < Q_2$.

We have subjected the PQ method based on double-compression to the blind steganalysis of [24]. This blind steganalysis uses 23 features derived from first-order (global histogram, individual histograms, and dual histograms) and higher-order statistics (spatial blockiness, co-occurrence matrices of coefficients from neighboring blocks, etc.) of DCT coefficients. The features are calibrated using the shifted/cropped/recompressed image first used in [25] for accurate estimation of secret message length. By using the calibrated features in this manner, one can significantly decrease image to image variations among features and vastly improve the detection sensitivity. Also, because the features are calculated directly from the DCT it is possible to directly draw conclusions about the impact of the embedding changes on detectability. As shown in [24], this detection scheme was able to reliably detect OutGuess [26] at embedding rates as low as 0.05 bpc and F5 [27] at 0.1 bpc. The Model based Steganography of [28] was also detected at full capacity of 0.4 bpc. Because, to the best knowledge of the authors, this detection is the only one that reliably detects all current state of the art steganographic techniques for JPEGs, we selected it as a benchmark for our tests as well.

bpc	F5	F5_111	OG	MB1	MB2	PQ
0.05	0.241	0.645	0.879	0.220	0.163	~ 0
0.1	0.539	0.922	0.993	0.415	0.310	0.048
0.2	0.956	0.996	0.991	0.704	0.570	0.098
0.4	1.000	1.000	U	0.938	0.824	0.174
0.6	1.000	1.000	U	0.983	U	U
0.8	1.000	1.000	U	0.992	U	U

Table 2 Detection reliability ρ for F5, F5 without matrix embedding (1,1,1), OutGuess 0.2 (OG), Model based Steganography without and with deblocking (MB1 and MB2, respectively), and the proposed Perturbed Quantization during double compression for different embedding rates (U = unachievable rate). All but the PQ algorithm, were tested with $Q = 80$. The PQ algorithm was tested with $Q_1 = 85$ and $Q_2 = 70$.

The Greenspun database of 1812 grayscale images (www.greenspun.com) was used for testing. The Fisher Linear Discriminant was trained on the set of 23 features for the first 1412 cover and fully embedded images. By cover images, we understand images that were subjected to a regular double compression with $Q_1 = 85$ and $Q_2 = 70$, while the stego images

were obtained by embedding a random message of length 0.4, 0.2, 0.1, and 0.05 bpc (bits per non-zero DCT coefficient of the stego image). The testing was done on the remaining set of 400 images in the database. On average, fully embedded images were able to accept approximately 0.48 bpc of the double-compressed image. As in [24], the detection was evaluated using the detection reliability ρ , which is the area between the ROC curve and the diagonal line in the ROC diagram (normalized so that $\rho = 1$ perfect detection, $\rho = 0$ no detection).

As can be seen from Table 2, the new algorithm significantly outperforms existing steganographic algorithms for JPEG images. Figure 4 shows ROC curves when testing for images fully embedded with PQ (on average 0.48 bpc).

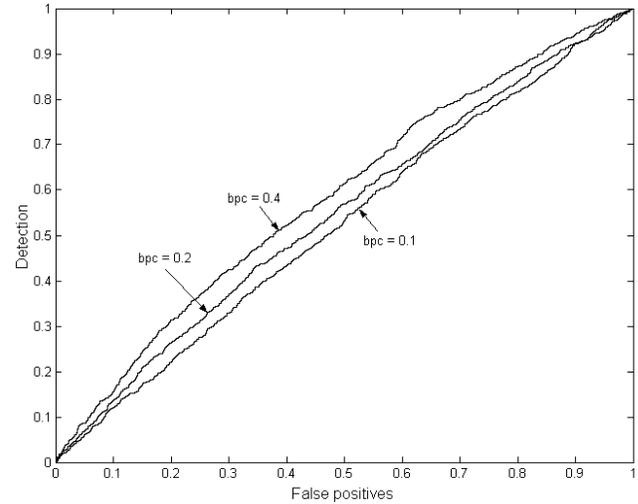


Figure 4 ROC for 1812 images embedded using PQ with $Q_1 = 85$ and $Q_2 = 70$ for the embedding rate 0.4, 0.2, and 0.1 bpc.

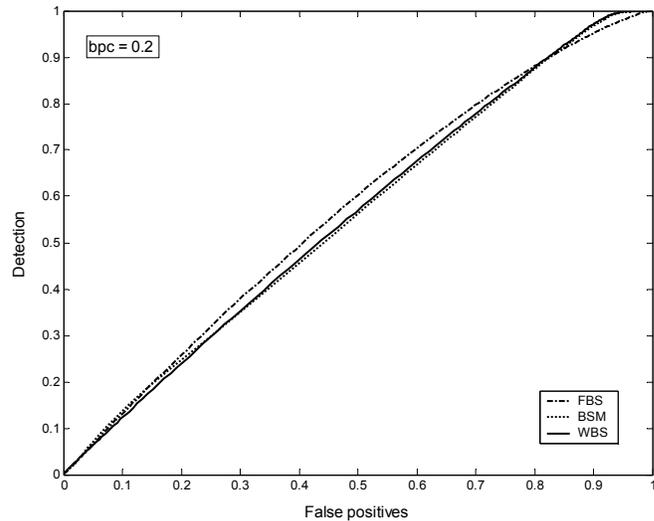


Figure 5 ROC for images embedded using PQ with $Q_1 \in [70, 85]$ and $Q_2 = 70$ for the embedding rate 0.2 bpc for three different steganalysis methods.

The security of perturbed quantization has recently been analyzed by other researchers. With kind permission of the authors in this paper we summarize the results that will appear in [29]. The PQ

was subjected to steganalysis on 207774 grayscale images with $3-7.5 \times 10^5$ pixels with the quality factor in the range 70–85. The tests were done with three blind steganalyzers that use features constructed in the wavelet domain (WBS) [30], features obtained from Binary Similarity Measures (BSM) calculated in the spatial domain [31], and the above 23 mentioned features calculated in the DCT domain (FBS). The second quality factor was set so that the quantization steps in the second quantization matrix were twice as big as in the first table (see the expression in the caption of Figure 3). This led to an average embedding capacity of 0.2 bpc. The results are shown in Figure 5. As can be seen, the PQ cannot be reliably detected at this embedding rate by any of the three classifiers.

6. CONCLUSIONS

In this paper, we use the wet paper code to develop new steganographic methodology for digital media called Perturbed Quantization. In Perturbed Quantization, the sender embeds a secret message while downgrading the cover object using some information-reducing operation that involves quantization, such as lossy compression, A/D conversion, downsampling, etc. The sender uses his knowledge of the *unprocessed* object and embeds data into those elements whose values are the most “uncertain” after the processing – they lie in the middle of quantization intervals. As this selection channel is based on information that is largely unavailable to the recipient, wet paper codes are applied to solve the problem of a non-shared selection channel.

We illustrate Perturbed Quantization on the example of recompressing a JPEG image with a lower quality factor. Blind steganalysis shows that Perturbed Quantization is significantly less detectable than existing steganographic methods for JPEG images while providing a relatively large capacity.

It might be perhaps feasible to develop attacks on PQ for some cover images by analyzing those areas in the image where more accurate prediction of the unquantized values is possible. For example, images containing large portions of blue sky or other uniform areas could be interpolated, recompressed on the same 8×8 grid, and the DCT quantized coefficients compared to the corresponding coefficients in the JPEG file. If this attack is, indeed, possible, the PQ would have to adopt more complex coefficient selection criteria that would be aware of the “predictability” of unquantized DCT coefficients from the stego image.

This predictability is directly related to the loss of information due to the information-reducing character of pre-processing. By quantifying this loss in information-theoretical terms and by assuming an appropriate model of the cover object, we might obtain bounds on the steganographic capacity of PQ.

7. ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under the research grant number F30602-02-2-0093. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U. S. Government. Special

thanks belong to Roman Tzschoppe and Dorin Hogeia for many useful discussions, and to Nasir Memon and Mehdi Kharrazi for providing their results on blind steganalysis of PQ.

8. REFERENCES

- [1] G.J. Simmons, “The Prisoners’ Problem and the Subliminal Channel”, *CRYPTO83 – Advances in Cryptology*, August 22–24, pp. 51–67, 1984.
- [2] F.A.P. Petitcolas and S. Katzenbeisser, editors, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House Books, January 2000.
- [3] R.J. Anderson and F.A.P. Petitcolas, “On the Limits of Steganography”, *IEEE Journal of Selected Areas in Communications*, Special Issue on Copyright and Privacy Protection, vol. 16(4), pp. 474–481, 1998.
- [4] C. Cachin, “An Information-Theoretic Model for Steganography”, in: D. Aucsmith (ed.): *Information Hiding, 2nd International Workshop*. Lecture Notes in Computer Science, vol. 1525, Springer-Verlag, New York, pp. 306–318, 1998.
- [5] J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, G. Wolf, “Modeling the Security of Steganographic Systems”, in: D. Aucsmith (ed.): *Information Hiding, 2nd International Workshop*, Lecture Notes in Computer Science, vol. 1525, Springer-Verlag, New York, pp. 344–354, 1998.
- [6] S. Katzenbeisser and F.A.P. Petitcolas, “Defining Security in Steganographic Systems”, *Proc. Electronic Imaging, SPIE, Security and Watermarking of Multimedia Contents IV*, vol. 4675, pp. 50–56, 2002.
- [7] R. Chandramouli, M. Kharrazi, and N. Memon, “Image Steganography and Steganalysis: Concepts and Practice”, in T. Kalker et al. (eds.): *Digital Watermarking, 2nd International Workshop*, Lecture Notes in Computer Science, vol. 2939, Springer-Verlag, New York Heidelberg, pp. 35–49, 2003.
- [8] E. Franz, “Steganography Preserving Statistical Properties”, in: F.A.P. Petitcolas (ed.): *Information Hiding, 5th International Workshop*. Lecture Notes in Computer Science, vol. 2578, Springer-Verlag, Berlin Heidelberg New York, pp. 278–294, 2002.
- [9] J. Fridrich and R. Du, “Secure Steganographic Methods for Palette Images”, in: A. Pfitzmann (ed.): *Information Hiding, 2nd International Workshop*. Lecture Notes in Computer Science, vol. 1768, Springer-Verlag, New York, pp. 47–60, 2000.
- [10] M. Karahan, U. Topkara, M. Atallah, C. Taskiran, E. Lin, E. Delp, “A Hierarchical Protocol for Increasing the Stealthiness of Steganographic Methods”, *Proc. ACM Multimedia Workshop*, Magdeburg, Germany, September 20–21, pp. 16–24, 2004.
- [11] A. Westfeld and R. Böhme, “Exploiting Preserved Statistics for Steganalysis”, in: J. Fridrich (ed.): *Information Hiding, 6th International Workshop*, Lecture Notes in Computer Science vol. 3200, Springer-Verlag, New York Heidelberg, pp. 82–96, 2005.

- [12] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk, "Writing on Wet Paper", *Proc. Electronic Imaging, SPIE, Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681, pp. 328–340, 2005.
- [13] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk, "Writing on Wet Paper" (journal version), to appear in *IEEE Trans. Sig. Proc.*, Supplement on Secure Media II, 2005.
- [14] A.V. Kuznetsov and B.S. Tsybakov, "Coding in a Memory with Defective Cells", *Probl. Inform. Transmission*, vol. 10, pp. 132–138, 1974.
- [15] C. Heegard and A. El-Gamal, "On the Capacity of Computer Memory with Defects," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 731–739, 1983.
- [16] R. Zamir, S. Shamai, U. Erez, "Nested Linear/Lattice Codes for Structured Multiterminal Binning", *IEEE Trans. Inf. Th.*, vol. 48(6), pp. 1250–1276, 2002.
- [17] G. Cohen, "Applications of coding theory to communication combinatorial problems. *Discrete Math.* vol. 83(2–3), pp. 237–248, 1990.
- [18] R.P. Brent, S. Gao, A.G.B. Lauder, "Random Krylov Spaces Over Finite Fields", *SIAM J. Discrete Math.* vol. 16(2), pp. 276–287, 2003.
- [19] M. Luby, "LT Codes", *Proc. The 43rd Annual IEEE Symposium on Foundations of Computer Science*, November 16–19, pp. 271–282, 2002.
- [20] E.R. Dougherty, *Random Processes for Image and Signal Processing*, SPIE PRESS Monograph Vol. PM44, 1998.
- [21] O. Ore and Y. Ore, *Number Theory and Its History*, Dover Publications, 1998.
- [22] J. Lukáš and J. Fridrich, "Estimation of Primary Quantization Matrix in Double Compressed JPEG Images", *Proc. of DFRWS 2003*, Cleveland, OH, August 5–8, 2003.
- [23] Ching-Yung Lin and Shih-Fu Chang, "Semi-Fragile Watermarking for Authenticating JPEG Visual Content", *Proc. Electronic Imaging, SPIE, Security and Watermarking of Multimedia Contents II*, vol. 3971, pp. 140–151, 2000.
- [24] J. Fridrich, "Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes", in: J. Fridrich (ed.): *Information Hiding, 6th International Workshop*, Lecture Notes in Computer Science vol. 3200, Springer-Verlag, New York Heidelberg, pp. 67–81, 2005.
- [25] J. Fridrich, M. Goljan, D. Hogeia, and D. Soukal, "Quantitative Steganalysis: Estimating Secret Message Length", *ACM Multimedia Systems Journal*. Special issue on Multimedia Security, 9(3), 288–302, 2003.
- [26] N. Provos, *Defending Against Statistical Steganalysis*, 10th USENIX Security Symposium. Washington, DC 2001.
- [27] A. Westfeld, "High Capacity Despite Better Steganalysis (F5–A Steganographic Algorithm)", in: I.S. Moskowitz (ed.): *Information Hiding. 4th International Workshop*, Lecture Notes in Computer Science, vol. 2137, Springer-Verlag, New York, pp. 289–302, 2001.
- [28] P. Sallee, "Model Based Steganography", in: T. Kalker, I.J. Cox, Yong Man Ro (Eds.), *Digital Watermarking. 2nd International Workshop*, Lecture Notes in Computer Science, Vol. 2939. Springer Verlag New York, pp. 154–167, 2004.
- [29] M. Kharrazi, H. T. Sencar, N. Memon, "Benchmarking Steganographic and Steganalytic Techniques", *Proc. Electronic Imaging, SPIE, Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681, pp. 252–263, 2005.
- [30] H. Farid and L. Siwei, "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines", in: F.A.P. Petitcolas (ed.): *Information Hiding. 5th International Workshop*. Lecture Notes in Computer Science, vol. 2578. Springer-Verlag, Berlin Heidelberg New York, pp. 340–354, 2002.
- [31] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using Image Quality Metrics", *Proc. Electronic Imaging, SPIE, Security and Watermarking of Multimedia Contents II*, vol. 4314, pp. 523–531, 2001.