Perturbed Quantization Steganography with Wet Paper Codes

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 introduce a new approach to passive-warden steganography in which the sender embeds the secret message into a certain subset of the cover object without having to share the selection channel with the recipient. An appropriate information-theoretical model for this communication is writing in memory with (a large number of) defective cells [1]. We describe a simple variable-rate random linear code for this channel (the "wet paper" code) and use it to develop a new steganographic methodology for digital media files – Perturbed Quantization. In Perturbed Quantization, the sender hides data while processing the cover object with an information-reducing operation, such as lossy compression, downsampling, A/D conversion, etc. The sender uses the cover object before processing as side information to confine the embedding changes to those elements of the processed cover object whose values are the most "uncertain". This informed-sender embedding and uninformed-recipient message extraction improves steganographic security because an attacker cannot easily determine from the processed stego object the location of embedding changes. Heuristic is presented and supported by blind steganalysis [2] that a specific case of Perturbed Quantization for JPEG images is significantly less detectable than current JPEG steganographic methods.

Categories and Subject Descriptors

E.4 Coding and Information Theory, I.4 Image processing and computer vision

General Terms: Algorithms, Security, Theory

Keywords: Adaptive, multimedia, quantizer, security, steganalysis, steganography

1. MOTIVATION

The primary goal of steganography is to build a statistically undetectable communication channel (the famous Prisoner Problem [3]). 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 [4,5], 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 [5–8]. 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.

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 [9–11], 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. A good example of a scheme where adaptive pixel selection in fact decreased its security is the recent surprising result of Westfeld [12].

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 recipient (and thus any 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 better steganography? The authors of this paper are not aware of any steganographic scheme that utilizes this side information, perhaps because it seems that the recipient would have to know the uncompressed image to read the message, which would not be practical. In this paper, we generalize this example and form a new steganographic method called "Perturbed Quantization". As explained in Section 2, this embedding method requires a coding technique for memories with a large number of defective cells. We call such codes "wet paper codes" because of the following metaphor that is highly relevant to steganography in general.

Imagine the 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 a memory with (*a large number of*) defective cells [1]. In Section 3, we describe a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM&Sec'04, September 20–21, 2004, Magdeburg, Germany.

Copyright 2004 ACM 1-58113-854-7/04/0009...$5.00.

simple variable-rate random linear code (the wet paper code) that gives the sender control over the embedding modifications and enables 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 code gives the sender complete freedom in choosing the dry pixels that will be used for embedding because the recipient does not need to (or even cannot in principle) determine the dry pixels from the stego image in order to read the message. This setup removes the above mentioned problem of adaptive steganography because it does not give an attacker a starting point for mounting an attack as when the selection rule is public. Also, the sender can now focus on the steganographic impact of the embedding changes without worrying whether or not the recipient will be able to read the message.

The proposed writing on wet paper shares some features with existing concepts in steganography and information theory, namely the selection channel [5], matrix embedding [13,14], and communication with side information [15]. The relationship between these concepts and writing on wet paper is commented upon throughout the paper in Sections 2.2, 3.1, and 3.2.

In Section 2, we describe the Perturbed Quantization steganography and propose several practical embedding scenarios. We then show that this steganographic communication requires codes for memories with a large number of defective cells. In Section 3, we propose a variable-rate random linear code for this channel that can be easily implemented even when the number of defective cells (wet pixels) is large. In the same section, we give a detailed pseudo-code for an encoder and decoder for practical applications. In Section 4, we describe a Perturbed Quantization steganographic technique 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 apply the blind steganalysis of [2] and compare the results with current state of the art JPEG steganographic algorithms. The paper is concluded 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, $i = 1, \ldots, n$, where n is the total number of DCT coefficients. 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 $\lfloor d \rfloor$ denotes the largest integer smaller than or equal to d.

Let $C = \{i_1, \ldots, i_k\}$ be the set of indices of all changeable coefficients. During compression, we will round changeable coefficients $d_j, j \in C$, up or down at our will and thus encode up to $k = |C|$ 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. In Section 3, we show how the sender can communicate on average $|C|$ bits to the recipient, who has no information about the set C.

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 shown in Section 3.6 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 reasons:

(1) The sender is using side information that is practically removed during quantization and is unavailable to the attacker. Indeed, it is in general impossible to reverse JPEG compression and obtain the uncompressed image or an approximation to the uncompressed image that would be good enough to enable the attacker to gain evidence that some coefficients were quantized "incorrectly".

(2) The sender can accept additional selection rule(s) to further decrease the probability of introducing detectable artifacts and thus improve the security. For example, the sender may avoid changing coefficients in those areas of the cover image where the attacker could predict the coefficient values with high certainty.

(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. The authors are currently working on a better justification of this heuristic statement using image models. It seems plausible that this heuristic can be, indeed, justified in a more exact manner by proving for a certain image model that the Perturbed Quantization is ε-secure in the Cachin's sense [6].

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. As discussed above, 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 an attack.

Let us assume that the cover image X is represented with a vector $x \in I^m$, where I is the range of its pixel/coefficient/color/index values depending on the format of X. For example, for an 8-bit grayscale image, $I=\{0, \ldots, 255\}$. The information-reducing process F will be modeled as a transformation

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

where J is the integer dynamic range of the downgraded image $Y = F(X)$ represented with an n-dimensional integer vector $y \in J^n$, $m \geq n$. 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" $T(X)$ will be denoted as U and represented using an n-dimensional vector $u\in \mathbf{R}^n$. We give several examples of image downgrading operations F that could be used for steganography based on PQ.

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

$$Q(u_i) = round(u_i). \quad (2)$$

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, …, 2^b–1}, i=0, …, m_1–1, j=0, …, m_2–1 into a $m_1\times m_2$ 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 $8\lceil m_1/8\rceil\times 8\lceil m_2/8\rceil$ matrix of real numbers u_{ij} in a block-by-block manner ($\lceil z\rceil$ denotes the smallest integer larger than or equal to z). In each 8×8 pixel block B^x, the corresponding block B^u in u_{ij} is DCT(B^x)./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).

2.2 Memory with defective cells

Continuing the description of Perturbed Quantization, the sender identifies the set of indices $C\subset\{1, …, n\}$ of pixels (or, in general, cover object *elements*) whose values u_j, $j\in C$, may be perturbed during quantization. The set C 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 X and 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

$$C=\{i\,|\,i\in\{1,…,n\},\ u_i\in[L+0.5–\varepsilon, L+0.5+\varepsilon] \text{ for some integer } L\}. \quad (3)$$

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 $u_j, j\in C$, to either L or L+1 and rounding all other elements u_i, $i\notin C$, 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 X and properties of the 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 b_i = {Parity(y_i)} be the sequence of parities[1] of elements from the processed cover object $Y = F(X)$. By perturbing the rounding process as described above, the sender can modify k bits b_j, $j\in C$, but cannot modify the remaining $n – k$ bits. The recipient does not know the set C. This is an example of a channel known as an n-bit memory with up to $n – k$ defective cells introduced in 1974 by Tsybakov et al. [1]. This channel is a special case of the Gelfand-Pinsker problem [15] of coding with side information. It is known that the capacity of this channel is k [17,21] and can be achieved, for example, using an algebraic coding scheme with the cosets of an erasure correction code as bins [18]. 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, however, the number of defective cells (wet pixels) 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 $k \sim 10^4$. To avoid the complexity associated with these codes when the number of defective cells is large, we describe a simple variable-rate random linear (wet paper) code that also enables the sender to communicate on average k bits and lends itself to practical applications in steganography. A significant advantage of this code is its flexibility and control it gives to the sender to choose which pixels should be modified, which further improves the security and minimizes the impact of embedding changes (Section 3.5).

3. Wet paper codes

3.1 Encoder

The proposed code can be viewed as a generalization of the selection channel [5] where one message bit is embedded as the parity of a group of 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. Now, if there are q elements in the group that can be changed, one can attempt to embed q message bits by forming q linearly independent linear combinations of element parities instead of just one sum.

Let us assume that the sender wants to communicate q bits $m = \{m_1, …, m_q\}^{\mathsf{T}}$. At this point, we assume that the recipient knows q. Later, we show how to modify the communication scenario to the case when the recipient does not know q. The sender and recipient agree on a secret stego key that is used to generate a pseudo-random binary matrix D of dimensions $q\times n$. The sender will round u_j, $j\in C$, obtaining the column vector y', so that the modified binary column vector b', b'_i = Parity(y_i'), i = 1, …, n, satisfies

$$Db' = m\,. \quad (4)$$

Thus, the sender needs to solve a system of linear equations in GF(2). The question of solvability of (4) is discussed in detail in Section 3.3. Note that the selection channel is a special case of (4) when D = [1, …, 1].

[1] The parity could be any function defined on J with range {0,1} such that Parity(k) = 1–Parity(k+1) for all $k\in J$. Thus, for J consisting of consecutive integers, only two parity functions are possible, $\text{Parity}_1(k)$ = LSB(k) or $\text{Parity}_2(k)$ = 1–LSB(k) (the shifted LSB). The Parity function could be the same for all elements or chosen randomly between Parity_1 or Parity_2 for each element based on a secret stego key.

3.2 Decoder

The sender sends the modified stego object $Y'=\{y_i'\}$ to the recipient. The decoding is very simple because the recipient first forms the vector b'_i = Parity(y_i') and then multiples Db' using the shared matrix D. The extracted message is simply $m = Db'$. The biggest computational load is on the sender's side who needs to solve (4).

The decoding mechanism is similar to that of matrix embedding [13,14] where the recipient also extracts the message bits by multiplying the parity vector by an appropriate code matrix. The difference is that in matrix embedding the sender's goal is to maximize the embedding rate utilizing the positions of the changes to convey information. While in matrix embedding any element can be modified, in writing on wet paper the set of elements that can be modified is pre-determined by the sender (or the cover object, or some randomness) beforehand and is different for different objects.

3.3 Average capacity

It will be advantageous to rewrite (4) to

$$Dv = m - Db \tag{5}$$

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

$$Hv = m - Db, \tag{6}$$

where H is a binary $q\times k$ submatrix of D and v is an unknown $k\times 1$ binary vector. This system has a solution for an arbitrary message m as long as $rank(H)\geq q$. The probability $P_{q,k}(s)$ that the rank of a random[2] $q\times k$ binary matrix is s, $s \leq \min(q,k)$, is [22, Lemma 4]

$$P_{q,k}(s) = 2^{s(q+k-s)-qk}\prod_{i=0}^{s-1}\frac{(1-2^{i-q})(1-2^{i-k})}{(1-2^{i-s})}\ . \tag{7}$$

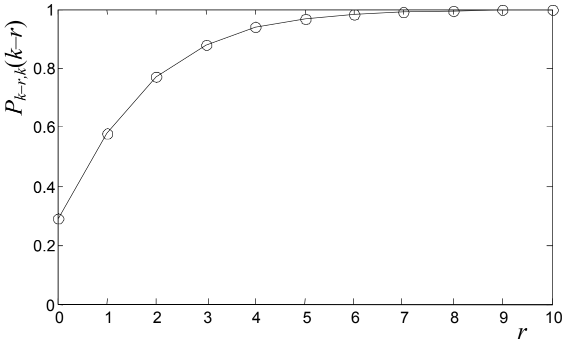

Figure 1. Probability that a random k–r by k binary matrix has rank k–r (for large k).

From (11) below, it can be shown that for a fixed large k, $P_{k-r,k}(k - r)$ quickly approaches 1 with increasing r (see Figure 1). This suggests that the expected maximal number of bits q that can be communicated is likely close to k, which is the theoretical upper bound. Next, we calculate the expected maximal number of bits that can be communicated using the wet paper code and show that it is approximately k. Due to the page limit for this paper, we only provide a rather sketchy proof of this statement leaving the details to our forthcoming paper [23].

For a given q-bit message, (6) may have a solution even when $rank(H) < q$ because each linearly dependent row in H is compatible with the corresponding bit on the right hand side with probability ½. Thus, the probability that one can communicate at least $k - r$ ($r\geq 0$) bits is

$$p_{\geq k-r} = \sum_{i=0}^{k-r}\frac{1}{2^i}P_{k-r,k}(k-r-i)\,, \tag{8}$$

while the probability that one can communicate at least $k + r$ ($r\geq 0$) bits is

$$p_{\geq k+r} = \sum_{i=0}^{k}\frac{1}{2^{r+i}}P_{k+r,k}(k-i)\,. \tag{9}$$

From (8–9), we calculate the expected maximum number $q_{\max}$ of bits communicated using k changeable pixels (the expected value taken over random messages and random matrices D)

$$q_{\max}(k) = \sum_{i=1}^{\infty} ip_{=i} = \sum_{i=1}^{\infty} i(p_{\geq i} - p_{\geq i+1}) = k, \tag{10}$$

where $p_{=i} = p_{\geq i} - p_{\geq i+1}$ is the probability that one can communicate exactly i bits. To prove that indeed $q_{\max}(k) = k$, we rewrite (7) using the function $\pi(i) = \prod_{j=1}^{i}\left(1-\frac{1}{2^j}\right), \pi(0) = 1$,

$$P_{q,k}(s) = 2^{s(q+k-s)-qk}\ \frac{\pi(q)\pi(k)}{\pi(s)\pi(q-s)\pi(k-s)}\ . \tag{11}$$

From Taylor expansion, we can easily show that $\pi(i) = \pi(\infty)(1+O(2^{-i}))$ for large i, where $\pi(\infty) = 0.288788\ldots$ by direct calculation. Because we are only interested in large values of k, q, and s (e.g., when $k = 100$ or larger), we can rewrite (11) using the asymptotic expression for $\pi(i)$ and substitute into (8) and (9). After some algebra and dropping the asymptotically small terms, we obtain for small $r\geq 0$

$$p_{\geq k-r} = \pi(\infty)\sum_{i=1}^{\infty}\frac{2^{-i^2-ri-i}}{\pi(i)\pi(i+r)}$$

and (12)

$$p_{\geq k+r} = \frac{\pi(\infty)}{2^r}\sum_{i=1}^{\infty}\frac{2^{-i^2-ri-i}}{\pi(i)\pi(i+r)}\ .$$

It is possible to prove by induction [23] that for large k the probabilities $p_{=i}$ are with a very high precision symmetrical about $i=k$: $p_{=k-r} = p_{=k+r}$ for $r = 1, 2, \ldots$ (see Figure 2). Consequently, from (10), $q_{\max}(k) \approx k$. *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 q. The sender and recipient can generate the matrix D in a row-by-row manner rather than generating it as a two-dimensional array of $q\times n$ bits. In this way, the sender can reserve the first few bits of the message m for a header of length $\lceil\log_2(n)\rceil$ bits to inform

[2] The probability of 0 and 1 in D is the same and equal to 1/2.

the recipient of the number of rows in D. The recipient first generates the first $\lceil \log_2(n) \rceil$ rows of D, multiplies them by the received vector b', and reads the header (the message length q). Then, he generates the rest of D, and reads the message m by multiplying Db'. Thus, under the assumption that the recipient has no information about either k or q, the sender can on average communicate $k - \log_2 n$ bits.

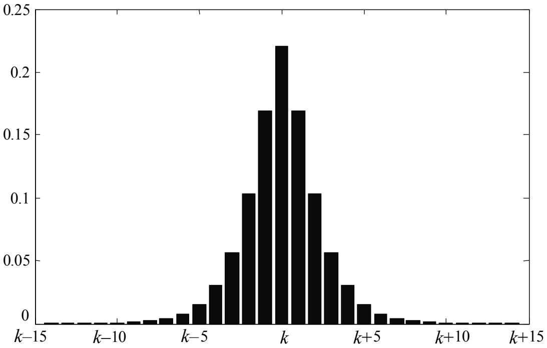

Figure 2. The probability $p_{=i}$.

3.4 Practical encoder implementation

The main complexity of this communication setup is on the side of the sender. The sender needs to solve q linear equations for k unknows in GF(2) (in binary arithmetic). Assuming that the maximal length message is sent, the complexity of Gaussian elimination for (6) is $O(k^3)$. For a medium size image with $n = 10^6$ pixels and the scenario in Example 3 with $\varepsilon = 0.1$, we have $k \sim 10^4$ for a typical 80% quality JPEG image. While solving a linear system with 10^4 unknowns using Gaussian elimination is doable on a PC, it may require several minutes of calculations, which is impractical for the user.

At this point, we stress that it is not possible to impose any specific structure on matrix H that would make the encoding easier because H is a submatrix of D obtained by selecting those columns from D that correspond to changeable pixels. Thus, different images will produce different matrices H even when D is kept the same.

Another possibility to solve (6) is to use more efficient solvers of linear systems. As shown in [24], $P_{k,k}(k) = 0.2889\ldots$ for random sparse matrices H with as few as $\log_2 k$ ones in each row. This fact opens up new possibilities in solving (6) significantly faster using techniques designed for sparse matrices. We investigated the Lanczos method [25] and the Wiedemann method [25]. Both methods have complexity proportional to $k(k+\omega k)(\log k)^c$, where ω is the average number of ones in each row of H and c is a small positive constant. Thus, they will be faster than Gaussian elimination for sufficiently large k. In our application, however, the matrix H is rectangular and may be singular, which complicates and slows down both methods. As a result, we did not find either method producing running times that would lead to a practical implementation (e.g., the order of at most few seconds for $n = 10^6$ and $k = 10^4$ to 10^5).

By far the best performance and most flexible method was obtained using structured Gaussian elimination by dividing the image into β pseudo-random disjoint subsets B_i and using the Gaussian elimination on each subset separately. This can bring down the computational requirements substantially because the complexity of Gaussian elimination will decrease by the factor of β^3 while the number of solvings increases β-times. This leads to performance improvement of β^2. A careful C++ implementation of the Gaussian elimination on a 2.2GHz PC (storing 32 bits of H and b as `int`) can solve a 1000×1000 system in about 0.02 seconds. Thus, for $\beta = 30$ subsets, the embedding of a maximal length message takes roughly 1 second. Dividing the image into subsets, however, brings new complications, such as the necessity to communicate the message length in each block, and thus leads to a slight decrease in embedding capacity (a few percent). Overall, the small decrease in capacity is well worth the significant improvement in speed. Below, we explain the embedding algorithm in detail.

Let us assume that the communicating parties know the range of typical values of the rate $r = k/n$, $r_1 \le r \le r_2$. If the range is unknown or r_2/r_1 is too large, the sender can modify the embedding algorithm below to communicate r [23] (not shown in this paper due to lack of space). The specific value of r will be influenced by the image content, the SR, and the transform T. To keep encoding time reasonably low, we desire approximately $k_{\text{avg}} \sim 250$ changeable pixels in each subset. We also require all subsets to be of almost the same size. Thus, we choose the number of sets $\beta = \lceil nr_2/k_{\text{avg}} \rceil$. The size n_i of each subset B_i will be $n_i \in \{\lfloor n/\beta \rfloor, \lceil n/\beta \rceil\}$ so that $n_1+n_2+\ldots+n_\beta = n$. Assuming the subsets are selected pseudo-randomly, there will be k_i changeable bits in each subset B_i, where k_i is a random variable with hypergeometrical distribution with mean k/β [23].

The number of message bits q_i embedded in each subset will be *allocated dynamically* during embedding by the sender (see the pseudo-code below). Without any loss of generality, we can assume that the image pixels are permuted using a pseudo-random permutation generated from a shared secret stego key. Then, the subsets B_i can simply be taken as subsets of n_i consecutive pixels, for example, in the row-by-row manner, and $b=(b^{(1)}, b^{(2)}, \ldots, b^{(\beta)})$, where $b^{(i)}$ is a vector of n_i parities of pixels in subset B_i. We are now ready to describe the encoder and decoder (see Figure 3).

Encoder

```
E0.  Using a PRNG, generate a random binary matrix $D$ with $\lceil n/\beta \rceil$ columns and sufficiently many rows
E1.  Determine the header size $h = \lceil \log_2(r_2 n/\beta) \rceil + 1$, $q = |m| + \beta h$
E2.  $b' \leftarrow b$, $i \leftarrow 1$
E3.  $q_i = \lceil k_i(q+10)/k \rceil$, $q_i = \min\{q_i, 2^h-1, |m|\}$, $m^{(i)} \leftarrow$ the next $q_i$ bits in $m$
E4.  Select the first $n_i$ columns and $q_i$ rows from $D$ and denote this submatrix $D^{(i)}$. Solve $q_i$ equations $H^{(i)}v = m^{(i)} - D^{(i)}b^{(i)}$ for $k_i$ unknowns $v$, where $H^{(i)}$ is a $q_i \times k_i$ submatrix of $D^{(i)}$ consisting of those columns of $D^{(i)}$ that correspond to changeable bits in $B_i$. If this system does not have a solution, the encoder decreases $q_i$ till a solution is found
E5.  According to the solution $v$, obtain the $i$-th segment $b'^{(i)}$ of the vector $b'$ by modifying or leaving $b^{(i)}$ unchanged
E6.  Binary encode $q_i$ using $h$ bits and append them to $m$
E7.  Remove the first $q_i$ bits from $m$
E8.  $q \leftarrow q - q_i$, $k \leftarrow k - k_i$, $i \leftarrow i + 1$
E9.  IF $i < \beta$ GOTO 3
E10. IF $i = \beta$, $q_\beta \leftarrow q$
E11. Binary encode $q_\beta$ using $h$ bits and prepend to $m$, $m^{(\beta)} \leftarrow m$
E12. Select the first $n_\beta$ columns and $q_\beta$ rows from $D$ and denote this submatrix $D^{(\beta)}$. Solve $q_\beta$ equations $H^{(\beta)}v = m^{(\beta)} - D^{(\beta)}b^{(\beta)}$ for
```

```
     $k_\beta$ unknowns $v$. If this system does not have a solution, exit and report failure to embed the message.
E13. According to the solution $v$, obtain the $\beta$-th segment $b'^{(\beta)}$ of the vector $b'$ by modifying or leaving $b^{(\beta)}$ unchanged
```

Decoder

```
D0. Using a PRNG, generate a random binary matrix $D$ with $\lceil n/\beta \rceil$ columns and sufficiently many rows
D1. Determine the header length $h = \lceil \log_2(r_2 n/\beta) \rceil + 1$
D2. $i \leftarrow \beta$
D3. Select the first $n_\beta$ columns and $h$ rows from $D$ and denote this submatrix $D_h$. Obtain $h$ bits as $D_h b'^{(\beta)}$ and decode as $q_\beta$
D4. Select the first $n_\beta$ columns and next $q_\beta - h$ rows from $D$ and denote this submatrix $D^{(\beta)}$. Obtain message bits $m = D^{(\beta)} b'^{(\beta)}$
D5. $i \leftarrow i - 1$
D6. Decode $q_i$ from the last $h$ bits of $m$ and remove the last $h$ bits from $m$
D7. Select the first $n_i$ columns and $q_i$ rows from $D$ and denote this submatrix $D^{(i)}$. Prepend $D^{(i)} b'^{(i)}$ to $m$, $m \leftarrow D^{(i)} b'^{(i)} \& m$
D8. IF $i > 1$ GOTO 5
D9. ELSE    $m$ is the extracted message
```

In Steps **E4** and **E12**, the sender forms an upper diagonal matrix from $H^{(i)}$ using Gaussian elimination, exchanging columns as needed to make sure that there will be 1's on the main diagonal. Once q_i rows are successfully processed, the sender sets the remaining values $v_i = 0$ for $i = q_i+1, \dots, k_i$ and calculates the unknowns v_i, $i = 1, \dots, q_i$. This will ensure that the embedding rate will always be close to 2 bits per change on average.

The encoder is allowed to decrease q_i whenever it cannot form an upper diagonal matrix (with ones on the diagonal) from $H^{(i)}$ using Gaussian elimination and by exchanging columns. The encoding process may fail in the last block because this is the only block in which the sender doesn't have the freedom to decrease q_β. To minimize the probability of this happening when q is close to k, the encoder is forced to embed slightly more bits in all other blocks than in the last one. This is the reason why the sender starts dividing the message bits with $q + 10$ rather than q (Step **E3**). Notice that one more bit is reserved for headers to cover a possibly larger k_i in a block than the expected value k/β. Because the header in each block has h bits, the message length in one block must not exceed $2^h - 1$ in Step **E3**.

The maximum number of bits that can be communicated using this algorithm is about

$$k - \beta h = k - \beta \times \lceil \log_2(r_2 k/\beta) \rceil. \quad (13)$$

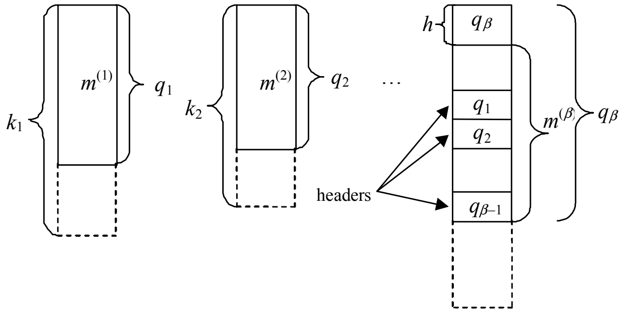

Figure 3. Placement of message bits and headers.

3.5 Minimizing the impact of embedding

When embedding a shorter than maximal message, in Steps E4 and E12 the sender will have freedom in choosing which unknowns v_i should be set to 0 and which will be determined by the Gaussian elimination. This freedom can be used to further minimize the impact of embedding. The SR will usually be formulated in quantitative terms and thus it will be possible to associate with each changeable sample x_i a numerical value $f(x_i)$ that somehow expresses its "fitness" to be included in the set of changeable samples. In Steps E4 and E12, when solving q_i equations $H^{(i)}v = m^{(i)} - D^{(i)}b^{(i)}$ for k_i unknowns v, the sender can solve for those unknowns v_i that correspond to samples with the largest fitness and set the remaining v_i's to zero. This way, the impact of embedding will be further minimized and the security of the scheme further improved.

A different way to minimize the impact of embedding is to minimize the number of embedding changes (maximize the embedding efficiency). With a fixed set of changeable pixels C, the problem of maximizing the rate is a binary vector quantization problem. To see this, we repeat that the sender needs to solve the system of q linear equations (6) $Hv = M - Db$ for k unknowns $v_1, \dots, v_k$. Also, recall that the non-zero elements of the vector v are the places where the sender needs to apply the perturbed quantizer. If $q < k$, the set of all solutions to (6) is of the form

$$v_0 + Ker(H)$$

where v_0 is one solution to (6) and $Ker(H)$ is the kernel of H formed by vectors x, such that $Hx = 0$. Minimizing the embedding distortion is equivalent to finding a vector $v = v_0 + x$ with the minimal Hamming weight. Thus, the sender needs to perform binary vector quantization, which is, however, known to be an NP complete problem. Never the less, there is a potential for improvement here even using suboptimal vector quantizers.

3.6 Perturbed quantizer

Assuming the SR is of the form (3), if the message bits form a random bit-stream, the act of embedding a message in the cover image X is well modeled with the probabilistic process $X \rightarrow Q_\varepsilon \circ T(X) = Y'$, where Q_ε is the perturbed quantizer

$$
\begin{aligned}
&Q_\varepsilon(z) = L \text{ for } L \le z < L+0.5-\varepsilon, \ (L \text{ is an integer})\\
&Q_\varepsilon(z) = L+1 \text{ for } L+0.5+\varepsilon \le z < L+1, \qquad (14)\\
&Q_\varepsilon(z) \in \{L, L+1\} \text{ with equal probability for } L+0.5-\varepsilon \le z < L+0.5+\varepsilon,
\end{aligned}
$$

and Y' is the stego image represented using an integer vector $y' \in J^m$. Note that $Q_\varepsilon = Q$ for $\varepsilon = 0$. The quantizers Q and Q_ε are identical with the exception of the interval $[L+0.5-\varepsilon, L+0.5+\varepsilon)$ where their output differs in 50% of cases. It can be easily shown that, assuming u is a random variable uniformly distributed on $[0, 1]$, the average quantization error $u - Q(u)$ introduced by the scalar quantizer (2) is 1/4, while for the perturbed quantizer it is $1/4+\varepsilon^2$. Thus, the difference between the average error of both quantizers is ε^2, which for $\varepsilon = 0.1$ is at least by one order of magnitude smaller than the average quantization error. Also, note that $-2\varepsilon \le |u - Q(u)| - |u - Q_\varepsilon(u)| \le 2\varepsilon$ for all u.

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 [2] 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 B^x (with integer pixel values in the range 0–255), the discrete cosine transform, $c = DCT(B^x)$, produces 64 DCT coefficients c_{ij}, $0 \le i, j \le 7$, which are then divided using the quantization matrix $q=(q_{ij})$ and rounded to integers using the quantizer (2)

$$\begin{aligned} c_{ij} &= \sum\nolimits_{k,l=0}^{7} a_{kl}(i,j) B_{kl}^{\;x} \\ d_{ij} &= c_{ij} / q_{ij} \qquad (15) \\ D_{ij} &= Q(d_{ij}). \end{aligned}$$

In (15), $a_{kl}(i,j)$ are the elements of the DCT transform matrix

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

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, the quantized DCT coefficients D_{ij} are multiplied by q_{ij}, inverse DCT transformed, and the result is rounded and clipped to a finite dynamic range obtaining the 8×8 pixel block B in the decompressed image

$$\begin{aligned} C_{ij} &= q_{ij} D_{ij} \\ B^{raw} &= DCT^{-1}(C) \qquad (17) \\ B &= [B^{raw}], \end{aligned}$$

where $[x] = Q(x)$ for $0 \le x \le 255$, $[x] = 0$ 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 X. Let B be an 8×8 block in X. Assuming that B has no pixels saturated at 0 or 255, from (17) we see that the quantization error $\xi_{ij} = B_{ij}^{raw} - B_{ij}$, $0 \le i, j \le 7$, satisfies $-0.5 \le \xi_{ij} \le 0.5$. Consequently,

$$DCT(B) = DCT(B^{raw}) - DCT(\xi) = C - \eta, \quad (18)$$

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

Modeling the quantization error ξ_{ij} as an i.i.d. noise uniformly distributed on the interval (–1/2, 1/2], we obtain

$$E(\eta_{ij}) = \sum\nolimits_{k,l=0}^{7} a_{kl}(i,j) E(\xi_{kl}) = 0,$$

$$E(\eta_{ij}{}^2) = \sum\nolimits_{k,l=0}^{7} a_{kl}{}^2(i,j) E(\xi_{kl}{}^2) + \sum\nolimits_{k,l=0}^{7} \sum\nolimits_{\substack{r,s=0 \\ (r,s)\ne(k,l)}}^{7} a_{kl} a_{rs} E(\xi_{kl} \xi_{rs}) = \frac{1}{12}$$

because $E(\xi_{ij}^2) = 1/12$ and $\sum\nolimits_{k,l=0}^{7} a_{kl}{}^2(i,j) = 1$ 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 [27]), the distribution of η_{ij} is approximately Gaussian $N(0,1/12)$.

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 4). As explained above, after decompression (17) and the second DCT transform (15), 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 4. 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 $Q_2 = 76$.

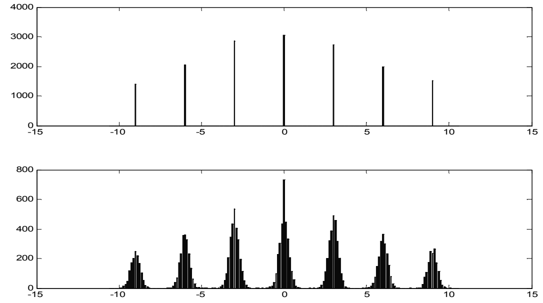

Figure 4 Top: histogram of values of the DCT coefficient C_{12} in the original 88% quality JPEG file (note that the values are multiples of the quantization step $q_{12}^{(1)} = 3$). Bottom: histogram of the same DCT coefficient c_{12} after decompressing the JPEG file to the spatial domain and DCT transforming.

From Figure 5, one can see that the peaks around the even multiples $2k$×3, k=0, 1, …, are quantized to $6k$, while the peaks around the odd multiples $(2k+1)$×3, k=0, 1, …, 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 $(2k+1)$×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

$$kq_{ij}^{(1)} = lq_{ij}^{(2)}+q_{ij}^{(2)}/2\ . \tag{19}$$

All integers k and l, l+1 that satisfy (19) 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* $g = \mathrm{GCD}(q_{ij}^{(1)}, q_{ij}^{(2)})$ *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*

$$k = (2m+1)\frac{q_{ij}^{(2)}}{2g}\ , m = \ldots, -2, -1, 0, 1, 2, \ldots. \tag{20}$$

Proof. The implication from left to right is trivial. Dividing (19) 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 [28], 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 (19) with $k = aq_{ij}^{(2)}/(2g)$ and $l = -bq_{ij}^{(2)}/(2g)$. To derive the formula (20), from (19) we have

$$k = \frac{(2l+1)q_{ij}^{(2)}}{2q_{ij}^{(1)}} = \frac{(2l+1)\dfrac{q_{ij}^{(2)}}{2g}}{\dfrac{q_{ij}^{(1)}}{g}}\ . \tag{21}$$

Because $\mathrm{GCD}(q_{ij}^{(1)}/g, q_{ij}^{(2)}/g) = 1$, 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. □

All contributing coefficients in the single compressed JPEG cover image form the set of changeable coefficients C. Theorem 1 can be used to calculate the cardinality of C. 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 $|C|$ is given by the following formula

$$|C| = \sum\nolimits_{i,j=0}^{7} \sum\nolimits_{k} u_{ij} h_{ij}\left((2k+1)\frac{q_{ij}^{(2)}}{2g}\right), \tag{22}$$

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

To show how $|C|$ depends on the quality factors Q_1 and Q_2, we evaluated (22) for all combinations of quality factors ranging from 50 to 95. The result was averaged over 20 test grayscale images and displayed in Figure 6. 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 5).

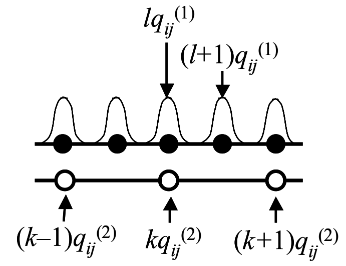

Figure 5 Example of a contributing multiple.

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 (22) with some reserve for the headers (13) and identifies the set C of changeable coefficients c_{ij} from the quantization matrices $q^{(1)}$ and $q^{(2)}$ using Theorem 1. From (19), the sender enforces that after the second JPEG compression, the quantized value D_{ij} (15) 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 encoding process as described in Section 3.4 while encoding the secret message.

To cast the embedding in the setup of Section 2.1, the transform $F = Q \circ T$ is composed of the decompression (17), the DCT transform (15), division by the second quantization matrix $q^{(2)}$, and the quantizer Q (2). Symbolically,

$$T(D_{ij}) = \mathrm{DCT}_{ij}([\mathrm{DCT}^{-1}(q_{ij}^{(1)}D_{ij})])/q_{ij}^{(2)}, \tag{23}$$

where $\mathrm{DCT}^{-1}(q_{ij}^{(1)}D_{ij})$ stands for the inverse DCT of the coefficient block to which D_{ij} belongs and $\mathrm{DCT}_{ij}(B)$ is the ij-th coefficient of the DCT of B.

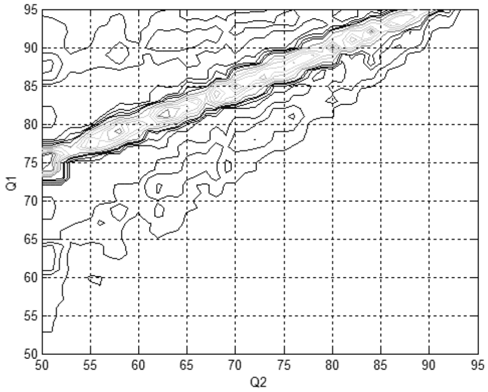

Figure 6 Embedding capacity expressed in bits per non-zero DCT coefficient (of the double-compressed image) averaged over 20 test images. Note the prominent ridge for quality factors satisfying $Q_2 = 2(Q_1 - 50)$.

5. STEGANALYSIS

In this section, we investigate the character of the embedding distortion and evaluate the security of the proposed algorithm using the approach described in [2].

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 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 [32] 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 [33]. 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 when $Q_1 < Q_2$ the double compression is *not* information-reducing.

We have subjected the PQ method based on double-compression to the blind steganalysis of [2]. 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 [29] 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 coefficients rather than from wavelet decomposition [30] or image quality metrics [31], it is possible to directly draw conclusions about the impact of the embedding changes on detectability. As shown in [2], this detection scheme was able to reliably detect OutGuess [35] at embedding rates as low as 0.05 bpc and F5 at 0.1 bpc. The Model based Steganography of [34] 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.

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 [2], 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 1, the new algorithm significantly outperforms existing steganographic algorithms for JPEG images. Figure 7 shows ROC curves when testing for images fully embedded with PQ (on average 0.48 bpc).

Table 1 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.

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

We close this section with some thoughts on the possibility of constructing a targeted attack on the proposed scheme. To construct a targeted attack, one would have to estimate the values of DCT coefficients prior to quantizing. While it is certainly possible to attempt to remove the JPEG quantization using smoothing techniques in the spatial domain, it will be extraordinarily difficult to estimate the unquantized coefficients with accuracy necessary to obtain sufficient evidence for presence of perturbed quantization. This is because in general one cannot reverse the loss of information that occurs during JPEG compression.

6. CONCLUSIONS

The main contributions of this paper are as follows. First of all, this paper reveals an important relationship between memories with defective cells [1] and steganography. The defective cells correspond to those cover object elements designated by the sender to be avoided for embedding and are *not* shared with the recipient. Because in steganography the number of defective cells could be quite large, we coin a new term for this steganographic channel – *writing on wet paper*. This is a metaphor for a steganographic channel in which the sender embeds message bits into a subset of elements of the cover object and communicates the message to the recipient, who does not have any information about the selection rule applied by the sender. If the selection rule is determined by side information available only to the sender but in principle unavailable to the recipient (and any attacker), this scenario provides improved steganographic security compared to schemes with a public selection rule [9–12].

Second, we propose a simple variable-rate random linear code (the wet paper code) for memories with a *large number* of defects and show how it can be applied for our steganographic channel. We prove that this code enables on average communication of k bits given k "dry" elements (n–k defective cells). The wet paper code lends itself to efficient practical implementations and offers flexibility and control to the sender over which cover object elements will be modified. This further minimizes the impact of embedding changes (Section 3.5).

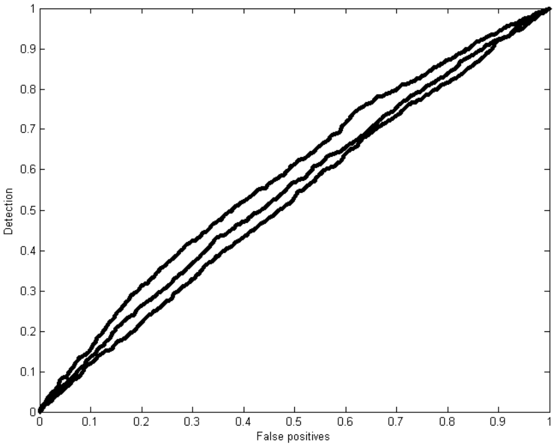

Figure 7 ROC for images embedded using PQ with Q_1 = 85 and Q_2 = 70 for the embedding rate 0.4, 0.2, and 0.1 bpc.

Third, using the wet paper code we 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 operations, such as lossy compression, A/D conversion, downsampling, etc. The sender uses his knowledge of the *unprocessed* object and embeds data into those pixels whose values are the most "uncertain" after the processing. We illustrate the methodology on the example of recompressing a JPEG image with a lower quality factor. Using heuristic arguments supported with blind steganalysis [2], we show that Perturbed Quantization is significantly less detectable than existing steganographic methods for JPEG images while providing a relatively large capacity.

Finally, we note that the writing on wet paper scenario and the proposed wet paper code can be thought of as a generalization of the selection channel [5]. The wet paper is also a special case of the general problem of communication with informed sender [15]. While the Costa's dirty paper code [16], which is another special case of [15], is relevant for watermarking [19,20], the wet paper is a suitable model for steganography.

There are other numerous applications of the wet paper code in steganography and general data embedding. For example, we name the removal of shrinkage in the F5 algorithm [13] and improving its embedding efficiency. Obviously, nullifying a DCT in F5 embedding coefficient will no longer be a problem for the decoder if the wet paper code is employed. Another application is constructing steganographic schemes that, besides the secret shared stego key, contain an element of true randomness and thus cannot be subjected to brute force stego key searches [36]. As the last application, we mention data hiding in binary images proposed by Wu [37]. In this application, the sender first identifies the set of "flippable" pixels that can be modified for embedding. Because this set of pixels is not shared with the recipient, Wu proposed block embedding combined with random shuffling. The block embedding however, leaves most of the flappable pixels unused and only a fraction of the embedding capacity is used. Because this problem exactly corresponds to writing on wet paper, the capacity of this data hiding method can be dramatically improved.

In the future, we plan to investigate in more detail the steganographic security of Perturbed Quantization. In particular, it seems plausible to prove its ε-security in the Cachin's sense [6] assuming an appropriate model of the cover object.

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 Petr Lisoněk, Pierre Moulin, and Ralf Koetter for many useful discussions, and to Hany Farid for providing the Greenspun image database.

8. REFERENCES

[1] A.V. Kuznetsov and B.S. Tsybakov, "Coding in a Memory with Defective Cells", *Probl. Inform. Transmission*, vol. 10, pp. 132–138, 1974.

[2] J. Fridrich, "Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes", *Proc. 6th Information Hiding Workshop*, Toronto, CA, May 23–35, 2004.

[3] G.J. Simmons, The Prisoners' Problem and the Subliminal Channel, *CRYPTO83* – Advances in Cryptology, August 22–24, pp. 51–67, 1984.

[4] F.A.P. Petitcolas and S. Katzenbeisser, editors, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House Books, January 2000.

[5] 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.

[6] C. Cachin, "An Information-Theoretic Model for Steganography", In: Aucsmith, D. (ed.): Information Hiding. 2nd International Workshop. Lecture Notes in Computer Science, Vol. 1525. Springer-Verlag, New York, pp. 306–318, 1998.

[7] J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, G. Wolf, "Modeling the Security of Steganographic Systems", In: Aucsmith, D. (ed.): Information Hiding. 2nd International Workshop. Lecture Notes in Computer Science, Vol. 1525. Springer-Verlag, New York, pp. 344–354, 1998.

[8] S. Katzenbeisser and F.A.P. Petitcolas, "Defining Security in Steganographic Systems", *SPIE Security and Watermarking of Multimedia Contents IV*, Vol. 4675, Electronic Imaging 2000, San Jose, CA, pp. 50–56, 2002.

[9] E. Franz, "Steganography Preserving Statistical Properties", In: Petitcolas, F.A.P. (ed.): Information Hiding. 5th International Workshop. Lecture Notes in Computer Science, Vol. 2578. Springer-Verlag, Berlin Heidelberg New York, pp. 278–294, 2002.

[10] J. Fridrich and R. Du, "Secure Steganographic Methods for Palette Images", In: Pfitzmann A. (ed.): Information Hiding. 2nd International Workshop. Lecture Notes in Computer Science, Vol. 1768, Springer-Verlag, New York, pp. 47–60. 2000.

[11] M. Karahan, U. Topkara, M. Atallah, C. Taskiran, E. Lin, E. Delp, "A Hierarchical Protocol for Increasing the Stealthiness of Steganographic Methods", to appear in Proc. ACM Multimediam Workshop, Magdeburg, Germany, September 20–21, 2004.

[12] A. Westfeld and R. Böhme, "Exploiting Preserved Statistics for Steganalysis", *Proc. 6th International Workshop on Information Hiding*, Toronto, Canada, May 23–25, 2004.

[13] A. Westfeld, "High Capacity Despite Better Steganalysis (F5–A Steganographic Algorithm)", In: Moskowitz, I.S. (eds.): 4th International Workshop on Information Hiding, LNCS, Vol. 2137. Springer-Verlag, New York, pp. 289–302, 2001.

[14] R. Crandall, "Some Notes on Steganography", posted on Steganography Mailing List, http://os.inf.tu-dresden.de/~westfeld/crandall.pdf, 1998.

[15] S.I. Gelfand and M.S. Pinsker, "Coding for channel with random parameters," *Probl. Pered. Inform.* (*Probl. Inform. Transm.*), vol. 9(1), pp. 19–31, 1980.

[16] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. IT-29(3), pp. 439–441, May 1983.

[17] 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.

[18] 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.

[19] P. Moulin and J. A. O'Sullivan, "Information-Theoretic Analysis of Information Hiding," *IEEE Trans. on Inf. Th*, vol. 49(3), pp. 563–593, March 2003.

[20] B. Chen and G. Wornell, Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding", *IEEE Trans. on Inf. Th.*, vol. 47(4), May 2001.

[21] G. Cohen, "Applications of coding theory to communication combinatorial problems. *Discrete Math.* **83** (2–3), pp. 237–248, 1990.

[22] R.P. Brent, S. Gao, A.G.B. Lauder, "Random Krylov Spaces Over Finite Fields", *SIAM J. Discrete Math.* **16**(2), pp. 276–287, 2003.

[23] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk, "Writing on Wet Paper", in preparation for IEEE Trans. Sig. Proc., Supplement on Secure Media II, 2004.

[24] C. Cooper, "On the Rank of Random Matrices", *Random Structures and Algorithms* **16**(2), pp. 209–232, 2000.

[25] B.A. LaMacchia and A.M. Odlyzko, "Solving Large Sparse Linear Systems over Finite Fields", In: Menezes, A.J., and Vanstone, S.A. (eds.): Advances in Cryptology – CRYPTO '90, Springer Verlag, Lecture Notes in Computer Science vol. 537, pp. 109–133, 1991.

[26] D.H. Wiedemann, "Solving Sparse Linear Equations Over Finite Fields", IEEE Transactions on Information Theory, IT–32(1), pp. 54–62, 1986.

[27] E.R. Dougherty, *Random Processes for Image and Signal Processing*, SPIE PRESS Monograph Vol. PM44, 1998.

[28] O. Ore and Y. Ore, *Number Theory and Its History*, Dover Publications, 1998.

[29] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, "Quantitative Steganalysis: Estimating Secret Message Length", *ACM Multimedia Systems Journal*. Special issue on Multimedia Security, **9**(3), 288–302, 2003.

[30] H. Farid and L. Siwei, "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines", In: Petitcolas, F.A.P. (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", SPIE Security and Watermarking of Multimedia Contents II, Electronic Imaging, San Jose, CA, Jan. 2001.

[32] 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.

[33] Ching-Yung Lin and Shih-Fu Chang, "Semi-Fragile Watermarking for Authenticating JPEG Visual Content", *SPIE Security and Watermarking of Multimedia Contents II*, Vol. 3971, Electronic Imaging 2000, San Jose, CA, pp. 140–151, 2000.

[34] P. Sallee, "Model Based Steganography", In: T. Kalker, I.J. Cox, Yong Man Ro (Eds.), International Workshop on Digital Watermarking, Lecture Notes in Computer Science, Vol. 2939. Springer Verlag New York, pp. 154–167, 2004.

[35] N. Provos, Defending Against Statistical Steganalysis, 10th USENIX Security Symposium. Washington, DC 2001.

[36] J. Fridrich, M. Goljan, and D. Soukal, "Searching for the Stego Key", *SPIE Security and Watermarking of Multimedia Contents VI*, Electronic Imaging 2004, San Jose, 2004.

[37] M. Wu, Tang E., and Liu B., "Data Hiding in Digital Binary Image", *Proc. Conf. on Multimedia & Expo* (ICME'00), New York City, 2000.