PERMUTATION CODES AND STEGANOGRAPHY

Félix Balado and David Haughton

School of Computer Science and Informatics, University College Dublin, Ireland

ABSTRACT

We show that Slepian's Variant I permutation codes implement first-order perfect steganography (i.e., histogram-preserving steganography). We give theoretical expressions for the embedding distortion, embedding rate and embedding efficiency of permutation codes in steganography, which demonstrate that these codes conform to prior analyses of the properties of capacity-achieving perfect stegosystems with a passive warden. We also propose a modification of adaptive arithmetic coding that near optimally implements permutation coding with a low complexity, confirming all our theoretical predictions. Finally we discuss how to control the embedding distortion. Permutation coding turns out to be akin to Sallee's model-based steganography, and to supersede both this method and LSB matching.

***Index Terms*—** Permutation coding, histogram preservation, arithmetic coding, model-based steganography, LSB matching

1. INTRODUCTION

First-order perfect steganography (histogram-preserving steganography) aims at empirically adhering to Cachin's criterion for undetectability [1]. It is a synonym with perfect steganography when the host elements are statistically independent. Although this assumption does not hold for real signals, a decorrelating energy-preserving invertible transform can always be applied before an optimum method that assumes statistical independence between symbols, as is usually done in the dual problem of lossless source coding (cf. Huffman coding). The integer KLT [2] seems a suitable choice. Here we study optimum first-order perfect steganography with a passive warden (i.e. no attack distortions), by delving deeper into its inextricable relationship with Slepian's permutation codes [3].

Notation and framework. Boldface lowercase Roman letters are column vectors. $\mathbf{1}$ and $\mathbf{0}$ are the all-ones vector and the null vector, respectively. $(\cdot)^t$ is the transpose operator. The 2-norm of $\mathbf{u}$ is $\|\mathbf{u}\| = \sqrt{\mathbf{u}^t\mathbf{u}}$. Capital Greek letters are matrices; $\mathrm{tr}\,\Pi$ is the trace of Π. Calligraphic letters are sets; $|\mathcal{X}|$ is the cardinality of $\mathcal{X}$. The indicator function is defined as $\mathbb{1}_{\{\mathrm{A}\}} = 1$ if event A is true, and zero otherwise. Logarithms are base 2 throughout the paper, unless noted otherwise. Uppercase Roman letters are random variables; $\mathrm{E}\{X\}$, $\mathrm{Var}\{X\}$ and $H(X)$ are the expectation, variance and entropy of X.

A host sequence is denoted by the discrete-valued n-vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^t \in \mathcal{V}^n$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_q\} \subset \mathbb{Z}$. We assume that $\mathbf{x} \neq v\mathbf{1}$, and also that $\mathbf{v} = [v_1, v_2, \ldots, v_q]^t$ gives the elements of $\mathcal{V}$ in increasing order, that is, $v_1 < v_2 < \cdots < v_q$. The histogram of $\mathbf{x}$ is a vector $\mathbf{h} = [h_1, h_2, \ldots, h_q]^t$ such that $h_k = \sum_{i=1}^n \mathbb{1}_{\{v_k = x_i\}}$; then $n = \mathbf{h}^t\mathbf{1}$. Let $\mathcal{S}_n$ be the group of all permutations of $\{1, 2, \ldots, n\}$. We denote a permutation $\boldsymbol{\sigma} \in \mathcal{S}_n$ by means of a vector $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_n]^t$ where $\sigma_i \in \{1, 2, \ldots, n\}$ and $\sigma_i \neq \sigma_j$ for all $i \neq j$. This vector can be used in turn to define a permutation matrix $\Pi_{\boldsymbol{\sigma}}$ with entries $(\Pi_{\boldsymbol{\sigma}})_{i,j} = \mathbb{1}_{\{\sigma_i = j\}}$. The rearrangement of $\mathbf{x}$ using $\boldsymbol{\sigma}$ is the vector $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\,\mathbf{x}$, for which $y_i = x_{\sigma_i}$ for $i = 1, 2, \ldots, n$. A special case is the rearrangement of $\mathbf{x}$ in nondecreasing order. This is obtained by means of a permutation $\overrightarrow{\boldsymbol{\sigma}}$ yielding $\overrightarrow{\mathbf{x}} = \Pi_{\overrightarrow{\boldsymbol{\sigma}}}\,\mathbf{x}$ such that $\overrightarrow{x}_1 \leq \overrightarrow{x}_2 \leq \cdots \leq \overrightarrow{x}_n$. Although $\overrightarrow{\mathbf{x}}$ is unique, $\overrightarrow{\boldsymbol{\sigma}}$ may not be so. The rearrangement of $\mathbf{x}$ in nonincreasing order is obtained as $\overleftarrow{\mathbf{x}} = \mathrm{J}\overrightarrow{\mathbf{x}}$, where J is the exchange matrix —a special permutation matrix with entries $(\mathrm{J})_{i,j} = \mathbb{1}_{\{j = n-i+1\}}$.

This work has been financially supported by Science Foundation Ireland under grant 09/RFP/CMS2212.

2. PERMUTATION CODES AND STEGANOGRAPHY

The fundamental observation in histogram-preserving steganography is that any information-carrying vector $\mathbf{y}$ that preserves the histogram of the host $\mathbf{x}$ has to be a rearrangement of $\mathbf{x}$. In other words, it must hold that the watermarked host is of the form $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\,\mathbf{x}$ for some permutation $\boldsymbol{\sigma} \in \mathcal{S}_n$, so that $\sum_{i=1}^n \mathbb{1}_{\{v_k = y_i\}} = \sum_{i=1}^n \mathbb{1}_{\{v_k = x_i\}}$ for all $k = 1, 2, \cdots, q$. If $\mathbf{x}$ can be rearranged into r different vectors $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(r)}$ then there are r histogram-preserving watermarks given by $\mathbf{w}^{(m)} = \mathbf{y}^{(m)} - \mathbf{x}$ for $m = 1, 2, \ldots, r$ (and hence at most r different messages); hereafter we will drop the superindex m whenever this is unambiguous from the context. The number r of rearrangements of $\mathbf{x}$ depends only on $\mathbf{h}$, and is given by the following multinomial coefficient:

$$r = \binom{n}{h_1, h_2, \cdots, h_q} = \frac{n!}{h_1!h_2!\cdots h_q!}. \tag{1}$$

In the remainder we will consider $\mathcal{S}_{\mathbf{x}} \subset \mathcal{S}_n$ to be an arbitrary set of permutations leading to the $r = |\mathcal{S}_{\mathbf{x}}|$ different rearrangements of $\mathbf{x}$.

According to the previous observations, first-order perfectly steganographic codes are the Variant I permutation codes first described by Slepian [3]. Any histogram-preserving steganographic strategy necessarily uses these codes, *or a subset of them*. Mittelholzer [4] was the first to consider Slepian's permutation modulation in steganography. He proved that the mutual information between the watermarked host and the embedded information is null when $\mathbf{y} = \mathbf{x} + \Pi_{\boldsymbol{\sigma}}\mathbf{k}$, with $\mathbf{k}$ a secret vector. However he did not investigate the histogram-preserving case $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\mathbf{x}$, that, in part, mirrors the use of permutation codes in channel/source coding [3, 5] —i.e. all codewords $\mathbf{y}$ are permutations of a base vector $\mathbf{x}$. In channel/source coding $\mathbf{x}$ is a design choice: for instance, in channel coding under a Gaussian i.i.d. channel, $\mathbf{x}$ is chosen to maximise the minimum pairwise distance between codewords because in this case minimum distance decoding implements maximum likelihood decoding [3]. However in steganography $\mathbf{x}$ is the host, and, as such, a fixed input parameter of the encoder. This fact conditions the two most relevant issues faced by permutation coding in steganography: encoder complexity —r can be very high, and cannot be fixed— and embedding distortion control —think of a random rearrangement of a real signal. In order to address these issues, we first undertake a theoretical analysis of permutation codes for steganography.

2.1. Embedding Distortion

Whether $\mathbf{x}$ is a real signal or its decorrelation, preserving its empirical distribution as propounded by Cachin's criterion is clearly not enough, as we also have to approximately preserve the semantics of the actual realisation $\mathbf{x}$. Given a codeword $\mathbf{y}$, a convenient way to measure its semantic closeness to $\mathbf{x}$ is by means of the squared Euclidean distance $\|\mathbf{y}-\mathbf{x}\|^2 = \|\mathbf{w}\|^2$. As $\|\mathbf{y}\| = \|\mathbf{x}\|$ with histogram preservation, this amount can be put as

$$\|\mathbf{w}\|^2 = 2\left(\|\mathbf{x}\|^2 - \mathbf{x}^t\mathbf{y}\right) = 2\left(\|\mathbf{x}\|^2 - \mathbf{x}^t\Pi_{\boldsymbol{\sigma}}\mathbf{x}\right) \tag{2}$$

for some $\boldsymbol{\sigma} \in \mathcal{S}_{\mathbf{x}}$. The average watermark power with equally likely messages $\overline{\|\mathbf{w}\|^2} \triangleq (1/r)\sum_{m=1}^{r}\|\mathbf{w}^{(m)}\|^2$ is therefore

$$\overline{\|\mathbf{w}\|^2} = 2\left(\|\mathbf{x}\|^2 - \frac{1}{r}\mathbf{x}^t\left(\sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\mathbf{x}}}\Pi_{\boldsymbol{\sigma}}\right)\mathbf{x}\right). \tag{3}$$

In order to develop (3) we will use the following identity:

$$\sum_{\boldsymbol{\sigma}\in\mathcal{S}_n}\Pi_{\boldsymbol{\sigma}} = (n-1)!\,\mathbf{1}\mathbf{1}^t. \tag{4}$$

To prove this result consider the number of $n\times n$ permutation matrices which have a one at any given entry. This is equivalent to fixing the row of the permutation matrix corresponding to that entry, and therefore there are $(n-1)!$ possibilities for the remaining $n-1$ rows. Since this holds for any entry, equation (4) follows. Now see that $(1/n!)\sum_{\boldsymbol{\sigma}\in\mathcal{S}_n}\Pi_{\boldsymbol{\sigma}}\mathbf{x} = (1/r)\sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\mathbf{x}}}\Pi_{\boldsymbol{\sigma}}\mathbf{x}$ because every different vector in the first sum is repeated $\prod_{k=1}^{q}h_k!$ times. Combining this equation with (4) and using $\mathbf{x}^t\mathbf{1}\mathbf{1}^t\mathbf{x} = (\mathbf{x}^t\mathbf{1})^2$, (3) becomes

$$\overline{\|\mathbf{w}\|^2} = 2\left(\|\mathbf{x}\|^2 - \frac{1}{n}(\mathbf{x}^t\mathbf{1})^2\right). \tag{5}$$

It is important to observe that $(1/n)\overline{\|\mathbf{w}\|^2} = 2\,s_{\mathbf{x}}^2$, where $s_{\mathbf{x}}^2$ is the biased sample variance of $\mathbf{x}$. Therefore this result is the deterministic counterpart of the probabilistic analysis of the average embedding distortion of capacity-achieving perfect stegosystems given by Comesaña and Pérez-González [6, page 17], who showed that $(1/n)\,\mathrm{E}\{\|\mathbf{W}\|^2\} = 2\,\mathrm{Var}\{X\}$ for those systems.

It is also desirable to upper bound the maximum power of a first-order perfectly steganographic watermark, $(\|\mathbf{w}\|^2)_{\max} \triangleq \max_{m\in\{1,2,\cdots,r\}}\|\mathbf{w}^{(m)}\|^2$. A histogram-preserving scheme may not use all r codewords, or else it may employ messages with nonuniform probabilities, and hence its corresponding average watermark power may differ from (5). Furthermore, $(\|\mathbf{w}\|^2)_{\max}$ is the worst-case scenario for a single histogram-preserving watermark. In order to obtain this quantity we will use the rearrangement inequality $\mathbf{x}^t\mathbf{u} \geq \overrightarrow{\mathbf{x}}^t\overleftarrow{\mathbf{u}}$ [7, chapter 10], which holds for any two n-vectors $\mathbf{x}$ and $\mathbf{u}$. Setting $\mathbf{u} = \mathbf{y}$ and observing that $\mathbf{y}$ is a rearrangement of $\mathbf{x}$ we have from (2) and the rearrangement inequality that

$$(\|\mathbf{w}\|^2)_{\max} = 2\left(\|\mathbf{x}\|^2 - \overrightarrow{\mathbf{x}}^t\overleftarrow{\mathbf{x}}\right). \tag{6}$$

Expressions (5) and (6) must be normalised to be meaningful across different hosts. The following figures of merit for the embedding distortion can be put forward: peak document-to-average watermark power ratio $\xi^* = n(2^b-1)^2/\overline{\|\mathbf{w}\|^2}$ (assuming that the host is represented with b bits/sample), document-to-average watermark power ratio $\xi \triangleq \|\mathbf{x}\|^2/\overline{\|\mathbf{w}\|^2}$ and document-to-worst-case watermark power ratio $\xi_{\min} \triangleq \|\mathbf{x}\|^2/(\|\mathbf{w}\|^2)_{\max}$, which are related as follows: $\xi^* \geq \xi \geq \xi_{\min}$. From geometrical considerations, it can also be shown that $\xi_{\min} \geq \xi/2$. Needless to say, high document-to-watermark power ratios are required for fidelity purposes.

2.2. Embedding Rate

The steganographic embedding rate associated to a permutation code is $\rho \triangleq (1/n)\log r$ bits/host element. Obviously this rate has to be optimum for first-order perfect steganography, for the reasons discussed at the start of this section. To see this from a probabilistic perspective, assume that Stirling's approximation $\log_e z! \approx z\log_e z - z$ (for large z) holds for all factorials in (1). Then the embedding rate can be informally approximated as $\rho \approx -\sum_{k=1}^{q}\frac{h_k}{n}\log\frac{h_k}{n}$ bits/host element. If X is a random variable with probability mass function $\mathbf{p} \triangleq (1/n)\mathbf{h}$ then $\rho \approx H(X)$. This approximation was mentioned already by Berger et al. [5] in the context of permutation coding, and it was first found by Brillouin [8]. More rigorously, r is the cardinality of the *type* $\mathbf{p}$ of the host (because all histogram-preserving codewords $\mathbf{y}$ must have the same empirical distribution as $\mathbf{x}$) and then $\rho \leq H(X)$ [9]. These probabilistic interpretations of ρ coincide with the findings about the achievable rate of perfect stegosystems arrived at by Sallee [10], considering lossless source coding, and by Comesaña and Pérez-González [6], departing from Gel'fand and Pinskers' formula.

2.3. Embedding Efficiency

The embedding efficiency ($\overline{\varepsilon}$) of a steganographic method is the average number of message bits embedded per host element change [11]. In order to undertake its computation we define an auxiliary $q\times n$ matrix Λ whose entries are $(\Lambda)_{k,i} = \mathbb{1}_{\{v_k=x_i\}}$, and we let $\Omega \triangleq \Lambda^t\Lambda$. Now, $\mathrm{tr}\,\Lambda\Pi_{\boldsymbol{\sigma}}\Lambda^t = \mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}}$ is the number of elements in $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\mathbf{x}$ unchanged with respect to $\mathbf{x}$. Hence, embedding the message associated to codeword $\mathbf{y}$ requires $n - \mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}}$ host element changes. To start with we will compute the average degree of host change, which is defined as $\overline{\nu} \triangleq (1/r)\sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\mathbf{x}}}(n - \mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}})/n$ when all messages are equally likely. To evaluate this expression observe that $(1/n!)\sum_{\boldsymbol{\sigma}\in\mathcal{S}_n}\mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}} = (1/r)\sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\mathbf{x}}}\mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}}$, because $\mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}}$ is constant across all permutations $\boldsymbol{\sigma}\in\mathcal{S}_n$ leading to the same rearrangement of $\mathbf{x}$. As the trace operator is linear, using again equation (4) and $\mathrm{tr}\,\Omega\mathbf{1}\mathbf{1}^t = \mathbf{1}^t\Omega\mathbf{1} = \|\Lambda\mathbf{1}\|^2 = \|\mathbf{h}\|^2$, it is straightforward to see that

$$\overline{\nu} = 1 - \frac{\|\mathbf{h}\|^2}{n^2} = 1 - \|\mathbf{p}\|^2. \tag{7}$$

Notice that $\overline{\nu}$, which only depends on the norm of the type of $\mathbf{x}$ and is bounded as $0 < \overline{\nu} \leq 1 - 1/q$, may be seen as an embedding distortion measure alternative to the figures of merit at the end of Section 2.1. We are now ready to address the computation of $\overline{\varepsilon}$. The embedding efficiency for the message encoded by $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\mathbf{x}$ is $\log r/(n - \mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}})$ bits/host element change. This amount is infinite for $\boldsymbol{\sigma}' \in \mathcal{S}_{\mathbf{x}}$ such that $\Pi_{\boldsymbol{\sigma}'}\mathbf{x} = \mathbf{x}$, which is why we adopt the criterion of taking the embedding efficiency to be zero in this case. Therefore, for equally likely messages, the average embedding efficiency is defined as $\overline{\varepsilon} \triangleq (1/r)\sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\mathbf{x}}\setminus\boldsymbol{\sigma}'}\log r/(n - \mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}})$. A useful lower bound on $\overline{\varepsilon}$ can be found by observing that it involves the harmonic mean of $r-1$ positive values, which is upper bounded by their arithmetic mean [7]. Then

$$\overline{\varepsilon} \geq \overline{\varepsilon}_l \triangleq n\rho\left(\frac{r-1}{r}\right)\left(\frac{1}{r-1}\sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\mathbf{x}}\setminus\boldsymbol{\sigma}'}(n - \mathrm{tr}\,\Omega\Pi_{\boldsymbol{\sigma}})\right)^{-1}. \tag{8}$$

Since the sum over $\boldsymbol{\sigma}\in\mathcal{S}_{\mathbf{x}}\setminus\boldsymbol{\sigma}'$ in (8) is equal to the same sum over $\boldsymbol{\sigma}\in\mathcal{S}_{\mathbf{x}}$, using the definition of $\overline{\nu}$ we have that

$$\overline{\varepsilon}_l = \frac{\rho\left(\frac{r-1}{r}\right)^2}{\overline{\nu}} \quad \text{bits/host element change.} \tag{9}$$

The approximation $(r-1)/r \approx 1$ is frequently needed to evaluate (9). In this case $\overline{\varepsilon} \gtrapprox \rho$, and $\overline{\varepsilon} \gtrapprox 2$ using inequality (19) from [12]

3. PRACTICAL ISSUES

Next, we discuss the two fundamental issues that arise in the practical implementation of permutation codes for steganography.

3.1. Near-Optimal Embedding Algorithm

Even for moderate n, the exponentially growing number of permutations precludes the implementation of a lookup table mapping messages to rearrangements of $\mathbf{x}$. However a low-complexity embedding algorithm can be devised by making the following observation: if there are r sequences with the same histogram $\mathbf{h}$ (for the same bins $\mathbf{v}$) then all of them can be uniquely represented with $nH(X) \approx \log r$ bits by means of optimum lossless source coding. Then optimum lossless source decoding of all different $(\log r)$-bits long messages should deliver all rearrangements of $\mathbf{x}$ (if informed by its statistics). Therefore first-order perfect steganography is dual of optimum lossless source coding, where embedding and decoding amount to decompression and compression, respectively. Sallee previously made this point for ϵ-secure steganography [10].

Arithmetic coding, being a near-optimal lossless source coding algorithm [13], can thus be used to implement permutation coding provided that a special adaptation procedure is used. In standard adaptive arithmetic coding one assumes an initial count of one for all symbols; after a symbol is encoded/decoded its count is increased. Here we assume that the initial symbol counts are the histogram values; after a symbol is encoded/decoded its count is decreased. Moreover we allow zero counts. In order to illustrate our adaptation strategy, assume that we wish to compress $\mathbf{x}$. Letting $\mathcal{I}^{(0)} \leftarrow [0,1)$ and $\mathbf{h}^{(0)} \leftarrow \mathbf{h}$, the i-th stage of our version of adaptive arithmetic coding (for $i = 1, 2, \cdots, n$) consists of dividing $\mathcal{I}^{(i-1)}$ into nonoverlapping right-open subintervals whose lengths are in proportion to the length of $\mathcal{I}^{(i-1)}$ according to the *nonzero* elements of $\mathbf{h}^{(i-1)}/(\mathbf{1}^t\mathbf{h}^{(i-1)})$. The subinterval for $h_k^{(i-1)} > 0$ is labelled as "v_k", and the one whose label v_l is equal to x_i is declared to be the new interval $\mathcal{I}^{(i)}$. The adaptation step consists of letting $h_l^{(i-1)} \leftarrow h_l^{(i-1)} - 1$ and declaring $\mathbf{h}^{(i)} \leftarrow \mathbf{h}^{(i-1)}$. Simply put, the contiguous most significant fractional bits shared by the binary representation of the endpoints of the final interval $\mathcal{I}^{(n)}$ constitute the compressed binary representation of $\mathbf{x}$. This compressed representation is roughly $nH(X) \approx \log r$ bits long, and then it can also be put as the most significant $\log r$ fractional bits of the decimal real value $(m'-1)/r \in [0,1)$, for some $m' \in \{1, 2, \cdots, r\}$. Decompressing m' requires doing the same subinterval division; in the i-th step, the label v_l of the subinterval where $(m'-1)/r$ lies is the decoded symbol x_i. In practice, finite-precision arithmetic has to be used to implement adaptive arithmetic coding for arbitrary n.

Now, to obtain the rearrangement $\mathbf{y} = e(\mathbf{x}, m)$ we simply carry out adaptive arithmetic decoding of m as described above. Clearly, the adaptation strategy guarantees that $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\mathbf{x}$ for some $\boldsymbol{\sigma} \in \mathcal{S}_{\mathbf{x}}$. This procedure can generate $2^{\lfloor \log r \rfloor} \approx r$ rearrangements of $\mathbf{x}$; for this reason, the empirical results obtained with this encoder closely follow all the theoretical expressions in Section 2. Decoding the message embedded in $\mathbf{y}$, that is, retrieving $m = d(\mathbf{y})$, entails carrying out adaptive arithmetic encoding of $\mathbf{y}$. Crucially, encoder and decoder share $\mathbf{h}$ precisely because permutation coding is implemented. The method may incorporate a symmetric secret key K (that is, $e_K(\cdot,\cdot)$ and $d_K(\cdot)$) by means of a shared permutation of $\mathbf{v}$. Finally, closely connected to this implementation of permutation coding are: 1) Jelinek's algorithm for Shannon-Fano-Elias coding used by Berger et al. for permutation coding of sources [5] —however this algorithm predates finite-precision arithmetic coding; and 2) the realisation by Howard and Vitter that arithmetic decoding can generate random variables from any desired distribution [13].

3.2. Embedding Distortion Control (Partitioning)

A permutation code based on $\mathbf{x}$ may not directly meet a preestablished constraint on the minimum value of $\underline{\xi}$. A low $\underline{\xi}$ implies that $\mathbf{y}$ is not likely to resemble $\mathbf{x}$. However $\underline{\xi}$ can be raised by restricting the codewords to a judiciously chosen subset from the ensemble of all histogram-preserving codewords as follows: 1) partition $\mathbf{x}$ into p disjoint subvectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p$ with lengths $n_1, n_2, \cdots, n_p$ such that $\sum_{j=1}^{p} n_j = n$; and 2) undertake permutation coding within each of these subvectors independently, that is, $\mathbf{y}_j = \Pi_{\boldsymbol{\sigma}_j}\mathbf{x}_j$ with $\boldsymbol{\sigma}_j \in \mathcal{S}_{\mathbf{x}_j}$ for $j = 1, 2, \cdots, p$. This strategy still preserves the histogram of $\mathbf{x}$, as trivially $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\mathbf{x}$ for some $\boldsymbol{\sigma} \in \mathcal{S}_{\mathbf{x}}$, but it decreases the number of embeddable messages. This number is now $r = \prod_{j=1}^{p} r_j$, where r_j is the multinomial coefficient associated to the histogram $\mathbf{h}_j$ of subvector $\mathbf{x}_j$; hence the theoretical embedding rate becomes $\rho = (1/n)\sum_{j=1}^{p} \log r_j$. The average and maximum watermark power with partitioning can be shown to be $\overline{\|\mathbf{w}\|^2} = 2\left(\|\mathbf{x}\|^2 - \sum_{j=1}^{p}(1/n_j)(\mathbf{x}_j^t\mathbf{1})^2\right)$ and $(\|\mathbf{w}\|^2)_{\max} = 2\left(\|\mathbf{x}\|^2 - \sum_{j=1}^{p}\overrightarrow{\mathbf{x}_j}^t\overleftarrow{\mathbf{x}_j}\right)$, respectively; $\underline{\xi}^*$, $\underline{\xi}$, and $\xi_{\min}$ follow from these expressions. Retracing the same steps as in Section 2.3, the average degree of host change can be seen to be $\overline{\nu} = \sum_{j=1}^{p}(n_j/n)\,\overline{\nu}_j$, where $\overline{\nu}_j = 1 - (\|\mathbf{h}_j\|/n_j)^2$, and a lower bound on $\overline{\varepsilon}$ is again (9) but using the expressions for r, ρ and $\overline{\nu}$ just given.

If encoder and decoder share a partitioning, then all predictions above can be achieved by using adaptive arithmetic decoding (encoding) within each subvector independently. Of particular interest are *histogram-induced* partitionings, in which the subvectors $\mathbf{v}_1, \cdots, \mathbf{v}_p$, that correspond to $\mathbf{h}_1, \cdots, \mathbf{h}_p$, are pairwise disjoint, and which, then, can be seen as induced by a partitioning of $\mathbf{v}$. The obvious choice is that encoder and decoder preagree a partitioning. This *static* partitioning approach is the least desirable, because its resulting performance will be dependent on $\mathbf{x}$, and not on any theoretical target; however, we show in Sections 4.1 and 4.2 that certain static partitionings suffice for permutation coding to outdo some relevant steganographic techniques. Interestingly, *adaptive* (target-driven) partitioning is also implementable by relying on histogram-induced partitionings: 1) encoder and decoder agree on a theoretical target, such as a minimum $\underline{\xi}$; 2) the encoder chooses the theoretically optimum histogram-induced partitioning when applied to $\mathbf{x}$, and uses it to produce $\mathbf{y}$; and 3) the decoder also chooses the theoretically optimum histogram-induced partitioning when applied to $\mathbf{y}$ (as if it were the host), and uses it to decode the embedded information from $\mathbf{y}$. If the optimum is unique, both parties will agree on it through this strategy. This is because the theoretical analysis crucially yields identical results when the host is either $\mathbf{x}$ or any rearrangement $\mathbf{y}$ for any tentative histogram-induced partitioning —even if $\mathbf{y}$ was not obtained through that tentative partitioning from $\mathbf{x}$. However optimum adaptive partitioning is a combinatorial optimisation problem of the *generalised assignment* class, which are NP-hard and beyond the scope of this paper: there are B_q possible histogram-induced partitionings, where B_q is the q-th Bell number, and, for instance, $B_{256} \sim 10^{373}$. Also, the optimum is not necessarily unique, but both parties can replicate a sequence of optimisation steps and choose the first suitable partitioning. A suboptimal example of such a sequence is partitioning $\mathbf{v}$ into partitions formed by $s = \lceil q/p \rceil$ adjacent bins, for increasingly greater p.

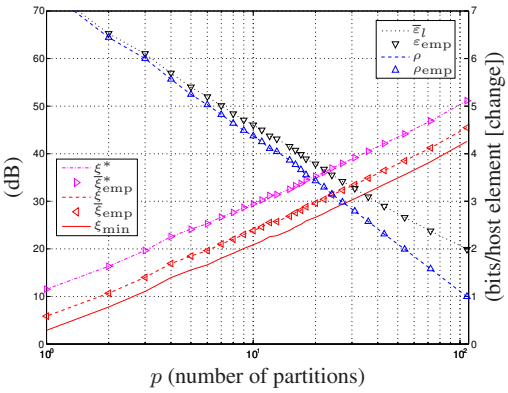

Fig. 1. Performance of permutation coding using adaptive partitioning. Lines are theoretical predictions and symbols are empirical results. Host: 512×512 grayscale Lena (8 bpp), spatial domain.

Fig. 2. 512×512 Lena (8 bpp) watermarked in the pixel domain using permutation coding ($\xi_{\text{emp}} = 41.22$ dB, $\xi^*_{\text{emp}} = 46.88$ dB, $\rho_{\text{emp}} = 1.58$ bpp, $\varepsilon_{\text{emp}} = 2.37$ bits/pixel change, $p = 72$, $s = 3$).

4. RESULTS

Figures 1 and 2 show the application of permutation coding using as the host $\mathbf{x}$ an arbitrary vector arrangement of the standard 512×512 grayscale Lena image with $b = 8$ bits/pixel (bpp), in the spatial domain. Note that $\mathbf{x}$ is obviously not decorrelated; the sole purpose of these plots is to illustrate the correctness of the analysis for a practical implementation of adaptive permutation coding. For each p a vector $\mathbf{y} = e(\mathbf{x}, m)$ is generated for a random m using the histogram partitioning strategy at the end of Section 3.2 and the encoder in Section 3.1; it is verified that $\mathbf{y}$ preserves the histogram of $\mathbf{x}$ and that the decoder correctly retrieves $m = d(\mathbf{y})$. The theoretical results are the ones in Section 3.2. The empirical results are $\xi_{\text{emp}} = \|\mathbf{x}\|^2/\|\mathbf{w}\|^2$ (document-to-watermark ratio), $\xi^*_{\text{emp}} = n(2^b - 1)^2/\|\mathbf{w}\|^2$ (peak signal-to-noise ratio, or PSNR), $\rho_{\text{emp}} = (1/n)\sum_{j=1}^{p} \lfloor \log r_j \rfloor$ bpp and $\varepsilon_{\text{emp}} = n\rho_{\text{emp}}/(\sum_{i=1}^{n} \mathbb{1}_{\{w_i \neq 0\}})$ bits/pixel change, where $\mathbf{w} = \mathbf{y} - \mathbf{x}$. If r_j cannot be computed exactly then $\log r_j$ is lower bounded using $\sqrt{2\pi z}(z/e)^z e^{(12z+1)^{-1}} < z! < \sqrt{2\pi z}(z/e)^z e^{(12z)^{-1}}$ [14], which is essential for the arithmetic decoder to work unambiguously. Remarkably, although the empirical results in Figure 1 represent single watermarks (i.e., they are *not* averages), they still accurately match the predictions involving averages. For $\underline{\xi}$ and $\underline{\xi}^*$ this stems from Chebyshev's inequality, which can be shown to yield $\Pr\{|\|\mathbf{W}\|^2 - \overline{\|\mathbf{w}\|^2}| \geq \delta\overline{\|\mathbf{w}\|^2}\} \leq 1/(\delta^2(n-1))$ assuming uniformly random permutations. Also, as discussed, $\xi_{\min} \geq \underline{\xi} - 3$ dB.

4.1. Comparison with LSB Matching (± 1 Steganography)

A static histogram partitioning grouping pairs of values from $\mathbf{v}$ which solely differ in their least significant bit (LSB) suffices for permutation coding (PC) to approximate the performance of Sharp's LSB matching ($\pm$1S) [15] which, however, is detectable using first-order statistics only [16]. A comparison for several 512×512 uncompressed images in the spatial domain is given below. $D(\mathbf{p}\|\mathbf{p}_\mathbf{y})$ is the relative entropy between $\mathbf{p}$ and the empirical distribution of $\mathbf{y}$.

	ρ_{emp} (ρ)		ε_{emp} ($\bar{\varepsilon}_l$)		ξ^*_{emp} ($\underline{\xi}^*$) [dB]		$D(\mathbf{p}\|\mathbf{p}_\mathbf{y})$	
	PC	$\pm$1S	PC	$\pm$1S	PC	$\pm$1S	PC	$\pm$1S
barb	0.99 (*0.99*)	1	1.99 (*1.99*)	2.01	51.15 (*51.15*)	51.15	0	7.0×10^{-4}
boat	0.99 (*0.99*)	1	1.99 (*1.99*)	2.00	51.17 (*51.18*)	51.14	0	7.4×10^{-3}
goldhill	0.99 (*0.99*)	1	1.99 (*1.99*)	2.00	51.14 (*51.14*)	51.15	0	1.6×10^{-3}
lena	0.99 (*0.99*)	1	1.99 (*1.99*)	1.99	51.14 (*51.15*)	51.14	0	5.7×10^{-4}
mandrill	0.99 (*0.99*)	1	1.99 (*1.99*)	2.00	51.15 (*51.14*)	51.13	0	5.8×10^{-4}

4.2. Comparison with Model-based Steganography

The defining difference between Sallee's model-based steganography (MB) [10] and permutation coding is that the former preserves a *theoretical model* of the host in a given domain, whereas the latter is instead domain-independent and preserves an *empirical model*. The nonadaptivity [10] or adaptivity of arithmetic coding is completely determined in both methods by their modelling approaches. Also, a probabilistic interpretation of the analysis in Section 2 shows that it extends and generalises the analysis in [10]. Below, we compare both methods for several 512×512 images JPEG-compressed with a quality factor of 80, using the static histogram partitioning "step size 2 embedding" from [10]: two adjacent bins per partition. We apply permutation coding separately to each frequency of the quantized coefficients in the 8×8 block DCT (all DC and null AC coefficients are skipped). $\overline{D}(\mathbf{p}\|\mathbf{p}_\mathbf{y})$ is the average relative entropy $D(\mathbf{p}\|\mathbf{p}_\mathbf{y})$ for the 64 empirical distributions of the frequencies.

	ρ_{emp} (ρ)		ε_{emp} ($\bar{\varepsilon}_l$)		ξ^*_{emp} ($\underline{\xi}^*$) [dB]		$\overline{D}(\mathbf{p}\|\mathbf{p}_\mathbf{y})$	
	PC	MB	PC	MB	PC	MB	PC	MB
barb	0.20 (*0.20*)	0.20	2.03 (*2.02*)	2.06	36.75 (*36.75*)	37.04	0	4.7×10^{-3}
boat	0.16 (*0.16*)	0.16	2.03 (*2.03*)	2.04	40.45 (*40.43*)	40.75	0	4.7×10^{-3}
goldhill	0.19 (*0.19*)	0.20	2.06 (*2.05*)	2.10	40.59 (*40.56*)	40.57	0	2.1×10^{-3}
lena	0.13 (*0.13*)	0.14	2.06 (*2.06*)	2.11	42.66 (*42.67*)	42.74	0	3.0×10^{-3}
mandrill	0.32 (*0.32*)	0.33	2.03 (*2.04*)	2.07	34.04 (*34.06*)	34.26	0	4.1×10^{-3}

Permutation coding essentially delivers the same performance while being simpler and more secure, since it is model-free, domain-independent and histogram-preserving (model-based steganography is detectable using first-order statistics only [17]), and more systematic and flexible, since a more complete analysis is available and partitioning can also be adaptive, rather than just static.

5. RELATION TO PRIOR WORK

This research was motivated by our work on information embedding in protein-coding DNA with codon bias preservation (a special case of first-order perfect steganography) [18, 19]. The only prior use of Slepian's permutation codes [3] in steganography was by Mittelholzer [4], but without histogram preservation. Our analysis shows that these codes comply with the results for capacity-achieving perfect stegosystems given by Comesaña and Pérez-González [6]. We have shown that, in practice, permutation coding supersedes Sharp's LSB matching [15] and Sallee's model-based steganography [10].

6. REFERENCES

[1] C. Cachin, "An information-theoretic model for steganography," in *Procs. of the 2nd Int. Workshop on Information Hiding*, ser. LNCS, vol. 1525. Portland, USA: Springer-Verlag, April 1998, pp. 306–318.

[2] H. Pengwei and Q. Shi, "Reversible integer KLT for progressive-to-lossless compression of multiple component images," in *Procs. of the 10th IEEE Int. Conf. on Image Processing (ICIP)*, vol. 1, Barcelona, Spain, September 2003, pp. 633–636.

[3] D. Slepian, "Permutation modulation," *Procs. of the IEEE*, vol. 53, no. 3, pp. 228–236, 1965.

[4] T. Mittelholzer, "An information-theoretic approach to steganography and watermarking," in *Procs. of the 3rd Int. Information Hiding Workshop*, ser. LNCS, vol. 1768. Dresden, Germany: Springer-Verlag, October 1999, pp. 1–16.

[5] T. Berger, F. Jelinek, and J. Wolf, "Permutation codes for sources," *IEEE Trans. on Information Theory*, vol. 18, no. 1, pp. 160 –169, January 1972.

[6] P. Comesaña and F. Pérez-González, "On the capacity of stegosystems," in *Procs. of the 9th ACM Workshop on Multimedia & Security*, Dallas, USA, September 2007, pp. 15–24.

[7] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*. Cambridge at the University Press, 1934.

[8] L. Brillouin, *Science and Information Theory*. Academic Press, 1962.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, 2006.

[10] P. Sallee, "Model-based steganography," in *Procs. of the 2nd Int. Workshop on Digital Watermarking (IWDW)*, Seoul, Korea, October 2003, pp. 154–167.

[11] A. Westfeld, "F5 – A steganographic algorithm," in *Procs. of the 4th Int. Information Hiding Workshop*, ser. LNCS, vol. 2137. Springer-Verlag, 2001, pp. 289–302.

[12] P. Harremoës and F. Topsøe, "Inequalities between entropy and index of coincidence derived from information diagrams," *IEEE Trans. on Information Theory*, vol. 47, no. 7, pp. 2944–2960, November 2001.

[13] P. Howard and J. Vitter, "Practical implementations of arithmetic coding," in *Image and Text Compression*, J. Storer, Ed. Kluwer Academic Publishers, 1992, pp. 85–112.

[14] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. John Wiley & Sons, 1968, vol. 1.

[15] T. Sharp, "An implementation of key-based digital signal steganography," in *Procs. of the 4th Information Hiding Workshop*, ser. LNCS, vol. 2137. Springer-Verlag, 2001, pp. 13–26.

[16] A. D. Ker, "Steganalysis of LSB matching in grayscale images," *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 441–444, June 2005.

[17] R. Böhme and A. Westfeld, "Breaking Cauchy model-based JPEG steganography with first order statistics," in *Procs. of the 9th European Symposium on Research in Computer Security (ESORICS)*, ser. LNCS. Springer-Verlag, 2004, vol. 3193, pp. 125–140.

[18] F. Balado and D. Haughton, "Gene tagging and the data hiding rate," in *Procs. of the 23rd IET Irish Signals and Systems Conference*, Maynooth, Ireland, June 2012.

[19] F. Balado, "Capacity of DNA data embedding under substitution mutations," *IEEE Trans. on Information Theory*, vol. 59, pp. 928–941, February 2013.