CERIAS Tech Report 2004-95
A Hierarchical Protocol for Increasing the Stealthiness of Steganographic Methods
by Mikhail J. Atallah
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

A Hierarchical Protocol for Increasing the Stealthiness of Steganographic Methods *

Mercan Topkara   Umut Topkara
Mikhail J. Atallah
Center for Education and Research in
Information Assurance, Purdue University
West Lafayette, Indiana, 47907 U.S.A.

Cuneyt Taskiran   Eugene Lin
Edward J. Delp
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, 47907 U.S.A.

ABSTRACT

We present a new protocol that works in conjunction with information hiding algorithms to systematically improve their stealthiness. Our protocol is designed to work with many digital object types including natural language text, software, images, audio, or streaming data. It utilizes a tree-structured hierarchical view of the cover object and determines regions where changes to the object for embedding message data would be easily revealed by an attacker, and are thus to be avoided by the embedding process.

The protocol requires the existence of a heuristic *detectability* metric which can be calculated over any region of the cover object and whose value correlates with the likelihood that a steganalysis algorithm would classify that region as one with embedded information. By judiciously spreading the effects of message-embedding over the whole object, the proposed protocol keeps the detectability of the cover object within allowable values at both fine and coarse scales of granularity. Our protocol provides a way to monitor and to control the effect of each operation on the object during message embedding.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Evaluation /methodology

General Terms

Algorithms, Security

Keywords

Steganography, steganalysis, statistical attacks, digital watermarking, upper bound on detectability, quad-tree structure

1. INTRODUCTION

The goal of steganography is to embed a message $\mathcal{M}$ in a cover object $\mathcal{C}$ in a covert manner such that the presence of the embedded $\mathcal{M}$ in the resulting stego-object $\mathcal{S}$ cannot be discovered by anyone except the intended recipient. Steganographic applications only require the flexibility to alter $\mathcal{C}$ in order to be able to embed the hidden information. For this reason any type of digital object can be potentially used as a cover. For example, images, audio, streaming data, software or natural language text have been used as cover objects.

Let Alice and Bob be two parties who exchange digital objects through a public communication channel. Alice and Bob would also like to exchange a secret message $\mathcal{M}$, however, they do not want the existence of this secret communication to be noticed by others. Alice and Bob do not want to achieve confidentiality through encryption, because the exchange of encrypted messages would reveal the existence of their secret communication. For this reason, they use a steganographic algorithm to embed $\mathcal{M}$ into a $\mathcal{C}$ to obtain a stego-object, $\mathcal{S}$, where $\mathcal{S} = (\mathcal{M}, \mathcal{C})$ and exchange $\mathcal{S}$ through the public communication channel.

The objective of the attacker Eva, is to construct a method for distinguishing stego-objects from unmodified objects with better accuracy than random guessing. Attack methods generally use statistical analysis to examine a suspicious object and search it for characteristics which may indicate that some information has been embedded in the object. For example, Eva might simply be looking for an unusual value of a characteristic that Alice has overlooked while modifying $\mathcal{C}$. Eva might also be looking for anomalies in the statistics of $\mathcal{S}$ that are different (e.g., finer) than the statistics Alice paid attention to when inserting the mark. Studies have shown that such statistical attacks are very successful on well-known image steganographic systems [17, 18, 5, 16, 8].

One way to defend against Eva's attacks is to inflict as little change to the document as possible [1, 21]. To this end, steganographic systems try to minimize changes in the cover object $\mathcal{C}$ when they are converted to corresponding

message-carrying regions in the stego object $\mathcal{S}$. Due to their statistical nature, some regions in the cover object will experience less change in their statistics after embedding. These message-carrying regions will be harder to identify for the attacker. Conversely, some regions will easily reveal their message-carrying characteristics. For example, in the case of an image steganography algorithm that uses random bit flipping, message-carrying regions will be easier to identify when the algorithm is applied to smooth regions compared to the case when it is applied to regions with high texture. In this case a region with natural noise is more suitable for message embedding than a smooth region.

This paper presents a general protocol for improving the stealthiness of a given steganographic algorithm by providing an efficient method to determine the most suitable regions to embed information. In our approach, we first partition the cover object $\mathcal{C}$ and impose a hierarchical structure $\mathcal{T}$ on it using this partitioning, where each node in $\mathcal{T}$ corresponds to a partition in the cover object $\mathcal{C}$. Then we use $\mathcal{T}$ both to monitor and to control the change in the statistics of the stego-object during the process of embedding the message, and to determine where the message bits are embedded.

Our protocol successfully masks the statistical effects caused by embedding both at fine and coarse levels from the attacker, since it allows constraints to be enforced on all levels of $\mathcal{T}$. Moreover the hierarchical nature of $\mathcal{T}$ allows us to impose an upper bound on the *detectability* in an arbitrary region even though the shape of this region may not be aligned with the boundaries that define the hierarchy.

For this paper we have chosen color images as cover objects. However, our protocol is applicable to other steganographic application domains, such as software, audio, streaming data, or natural language watermarking.

The paper is organized as follows: In Section 2 a brief overview of related work in steganography is given. Section 3 describes our protocol in detail. Section 4 discusses the experiments we have performed and presents results. Our conclusion are in Section 5.

2. PREVIOUS WORK IN STATISTICAL ATTACKS AND COUNTERMEASURES

Steganalysis is the study of methods and techniques to detect and extract hidden data in stego-objects that are created using steganographic techniques. These techniques generally introduce some amount of distortion in the stego-object during message embedding, even though this distortion may not easily be detected by a human observer. Steganalysis methods aim to exploit this fact by detecting statistical effects caused by the distortion to distinguish between cover objects and stego-objects. The challenge of designing a steganographic technique is to introduce the distortion in such a way as to minimize its statistical detectability by steganalysis. One approach, which was taken by early steganographic methods, was to try to minimize the detectability of data hiding by introducing as little distortion as possible during embedding. However, as pointed out by Fridrich and Goljan [7], recent advances in steganalysis have shown that this approach does not guarantee robustness against steganalysis, evidenced by the fact that least significant bit (LSB) embedding can successfully be attacked even for very short message lengths. This is due to the fact that LSB embedding introduces unnatural statistical artifacts that can easily be detected.

One of the first practical works on robustness against statistical attacks was [17], which introduced a statistical attack on stego-documents. This attack is based on the chi-square test, where the estimated color histogram distribution is compared with its observed values. Then the chi-square value, which shows the deviation from the expected values, is used to estimate of the probability that a given image has information embedded in it.

Provos [18] proposed a generalized chi-square attack that is capable of detecting more subtle changes in stego-documents. He introduced two methods for decreasing the distortion of the embedding process and for defending against generalized chi-square attack. A pseudo-random number generator is used to create multiple groups of bit selection for embedding. The selection that causes the fewest changes to the cover document is used for embedding. Later, error correction is applied to compensate for detectability caused by the embedding process. Provos incorporated these ideas in his steganographic system, *Outguess*, that embeds bits in the LSBs of DCT coefficients for JPEG images. He used a two-pass algorithm, where bits are embedded in the first pass and changes are made to coefficients in the second pass to match the histogram of DCT coefficients of the stego-image with that of the cover image. Since chi-square attacks rely on the first order statistics of the image, this makes the *Outguess* system immune to such attacks.

Westfeld, in his steganographic system *F5* [20], decrements the DCT coefficient's absolute values instead of overwriting the LSBs, in order to defend against chi-square test proposed in [17]. *F5* also uses matrix encoding to restrict the necessary changes on the cover object to embed the message. Matrix encoding helps to improve embedding efficiency significantly. Embedding efficiency is the ratio of embedding rate and necessary changes per message bit. Besides these, message bits are distributed over the whole cover image using permutative straddling.

Recently a number of algorithms that successfully attack the sophisticated steganographic systems were proposed. Fridrich et al. discuss a general methodology for developing attacks on steganographic systems using the JPEG image format, which is also effective for the *Outguess* and *F5* system [10]. Their approach is based on the assumption that there is a macroscopic quantity that predictably changes with the length of the embedded secret message for a given embedding method. Lyu and Farid [16] propose an attack that universally works for any steganographic system using images. It is based on higher-order statistical models of natural images, where use is made of a wavelet-like decomposition to model images and train a classifier with this model. This classifier is then used for classifying images as a cover image or a stego-image.

Another approach that tries to maintain image statistics after embedding is [6] where the embedding process is modeled as a Markov source and the required distribution of the embedding over the stego document to make it stealthy is determined.

Sallee [19] proposed an information-theoretic method for both steganography and steganalysis. A statistical model of the cover media is used to estimate $\hat{P}_{X_\beta | X_\alpha}(X_\beta | X_\alpha = x_\alpha)$ where x_β is the part of the cover object that is used for embedding and x_α is the remaining part which is unperturbed.

Then this model is used to select the value x'_β that conveys the intended secret message and is also distributed according to estimated $\hat{P}_{x_\beta|x_\alpha}$. This steganoghraphy method works for any type of cover media. Moreover, if this system is used, capacity of a cover medium can be measured using the entropy of the conditional distribution $\hat{P}_{x_\beta|x_\alpha}$ for a given x_α.

For in-depth discussion of other work on steganalysis and steganographic techniques we refer the reader to [8] and [13].

3. GENERAL FRAMEWORK

We define a protocol that can be used in conjunction with any embedding algorithm to control and improve the algorithm's stealthiness. We only require that a partitioning of the document is possible and that for any region a quantifiable measure, $d()$, that we denote as the *detectability* of the region, is defined to measure the likelihood that any steganalysis algorithm would classify that region as one with embedded information. However, this measure is hard to derive in practice. Therefore, we use a metric based on the degree the statistics of the region deviate from aggregate behavior of similar regions in a collection. For example, the detectability of an image block may be defined as the distance of the statistics of the block from the estimated statistics obtained for that block using an image model trained on the image or on a collection of related training images.

In the following subsections we discuss the properties of the hierarchical representation. We describe the details of the hierarchical representation in Section 3.1, and its advantages in Section 3.2. We conclude in Section 3.4 with a proof on the upper bound of detectability caused when the hierarchical representation is used during embedding.

3.1 Hierarchical Representation of the Cover-Document

In our approach the cover document is partitioned into blocks and a hierarchical structure is imposed on the document using this partitioning. This hierarchical structure is used to update the statistical properties of the document during embedding. Once this information is available, it can as well be used to efficiently manage the computational complexity of the process of choosing the suitable regions to embed information. More significantly, if the detectability caused by embedding is kept below a threshold at each node in the hierarchical representation, then we are guaranteed an upper bound on the detectability of any arbitrary region of interest in the object.

Let $\mathcal{T}$ be a tree used to represent the cover document $\mathcal{C}$. Each node N_i in this tree corresponds to a block in the partition of $\mathcal{C}$, denoted by $R(N_i)$, as illustrated in Figure 1. We use $\mathbf{T}(N_i)$ to refer to the vector of values that contain statistical information about block $R(N_i)$. The height of the subtree rooted at N_i is $h(N_i)$. The parent and the set of child nodes of N_i are denoted by parent(N_i) and children(N_i).

The nodes for which $h(N_i) = 0$ in $\mathcal{T}$ are called leaf nodes. If N_i is a leaf node, then we refer to $R(N_i)$ as an *elementary block*. n is the number of elementary blocks, which is equal to the number of leaf nodes in $\mathcal{T}$. The elementary blocks may correspond to paragraphs in natural language text, where we can perform either syntactic or semantic analysis of sentences [2] as well as text formatting analysis [3]. In software watermarking these elementary blocks might correspond to control flow blocks, whereas in images they could be blocks of pixels or regions of interest.
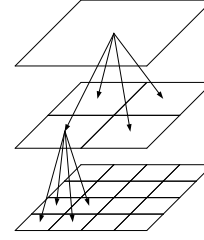
Figure 1: Hierarchical representation in the form of a quad-tree for a two-dimensional stego-document. Lower levels of the tree correspond to finer partitioning of the cover object.

For a given message $\mathcal{M}$ and a cover object $\mathcal{C}$, the *embedding algorithm* $f(\mathcal{M}, \mathcal{C})$ produces the stego-object, $\mathcal{S}$. We assume that f embeds each bit of the message, M_j, by performing one or more transformations on a block of $\mathcal{C}$. For example, the transformation could be the flipping of least significant bits in an image or the changing of active sentences into passive sentences in text. This transformation is called an *embedding operation*. More precisely, the embedding operation $G(M_j, R(N_i))$ takes the jth bit of $\mathcal{M}$, embeds it into the region $R(N_i)$ of $\mathcal{C}$ and produces $R'(N_i)$ of $\mathcal{S}$.

Depending on the structure of $\mathcal{C}$, $\mathcal{T}$ can be implemented as a binary tree, a quad-tree, or some other tree structure that need not have a fixed branching factor. $\mathcal{T}$ is formed such that $\mathbf{T}(N_i)$ may be obtained from $\sum_{\mathbf{v} \in \text{children}(N_i)} \mathbf{T}(\mathbf{v})$. We can reflect the statistical effects of $G(M_j, R(N_i))$ on $\mathcal{C}$ at leaf-level, upward, to all ancestor nodes of N_i in $O(height(\mathcal{T})$, which is $O(\log n)$ time.

3.2 Advantages of the Hierarchical Representation

Using the hierarchical representation in conjunction with an embedding algorithm provides the following advantages:

- A structured view of the statistical properties of the document is obtained for different resolutions, which will point out the *hot-spots*, which are the regions where the local statistics have anomalies compared to the global statistics of the document.

- It is possible to efficiently keep track of the changes in the statistics of the cover object after each embedding step. This is provided by reflecting the updates in statistics to higher levels in the hierarchical representation, which requires only $O(\log n)$ updates. n is the number of *elementary blocks*.

- Our protocol can set an upper bound on the detectability of arbitrary regions in the cover object if we preserve a threshold on detectability values at each level of the hierarchy. Section 3.4 contains a derivation of this upper bound.

- We can efficiently query document statistics. During the embedding process, some steganographic algorithms try to find the most suitable regions to embed information, as well as regions that require compensation for damage to the detectability incurred during

information embedding. In the hierarchical representation only the statistics on the path to the root are relevant. Whenever we detect an anomaly in statistics of regions on this path, we will be able to focus on one subtree for corrections, whose root stands out with an abnormal value. Siblings will cooperate in "fixing" the abnormality in their parent's statistics in this process of correction.

One drawback of using a pre-computed detectability metric or model of the cover medium, is that it does not keep track of the document statistics that change during embedding, which may affect the detectability. This may cause the algorithm to incur detectability that is larger than what was initially quantified by the cost metric. Another drawback is that there is no mechanism for backtracking from a change made in the document in favor of a better embedding option that appears later during embedding, which may cause sub-optimal embedding performance. Our protocol, on the other hand, dynamically updates document statistics by monitoring statistical properties of candidate embedding regions using the hierarchical structure on-the-fly during embedding. Stealthiness is achieved through an efficient representation of the embedding costs, and it allows the embedding system to avoid regions whose use might result in poor embedding performance.

If our protocol is used in conjunction with error correction, then making only one pass through the stego-document is enough. Contrast this with steganographic methods like *Outguess* [18], that try to preserve the statistics of the cover image through a two-pass approach. In the first pass, message data is embedded into regions which are found to be suitable using a static detectability metric. In the second pass additional non-embedding changes are made to compensate for the changes in the statistical properties of the object introduced in the first pass.

3.3 The Protocol

In this section we will describe the protocol that ensures that the detectability measure for a region, $d(R(N_i))$ after applying $G(M_j, R(N_i))$ stays below a threshold τ. This will allow our protocol to limit the increase in detectability introduced by the embedding algorithm, thereby increasing its stealthiness. An upper-bound on the detectability is derived in the next section.

For each node we define a binary-valued function $S(N_i)$ which we will refer to as the *suitability function*. $S(N_i) = 1$ if embedding any bit from $\mathcal{M}$ in N_i will not increase $d(R(N_i))$ beyond τ, i.e. $d(G(M_j, R(N_i))) < \tau$. We also keep track of whether a message bit was embedded in $R(N_i)$, in indicator $Z(N_i)$. At each step during the embedding N^* is the suitable node selected for the embedding operation.

Let $D(\mathbf{T}(N_i))$ be a function that returns the detectability value for node N_i given the statistics, $\mathbf{T}(N_i)$. $d(G(b, R(N_i)))$ is the detectability measure after applying the embedding operation over the region $R(N_i)$, where b is the part of the message that can be embedded in $R(N_i)$.

INITIALIZATION PHASE
   for each N_i in $\mathcal{T}$ in a bottom-up manner
      do $Z(N_i) \leftarrow 0$
         $S(N_i) \leftarrow 1$
         if N_i is a leaf node
            perform analysis on $R(N_i)$ to obtain $\mathbf{T}(N_i)$

      else
         $\mathbf{T}(N_i) \leftarrow \Sigma_{\mathbf{v} \in \text{children}(N_i)} \mathbf{T}(\mathbf{v})$
       $d(R(N_i)) \leftarrow D(\mathbf{T}(N_i))$
   for each N_i in $\mathcal{T}$ in a top-down manner
      do if $d(G(b, R(N_i))) > \tau$
         then $S(N_i) \leftarrow 0$
            for each N_j in the subtree with root N_i
               do $S(N_j) \leftarrow 0$

EMBEDDING & DYNAMIC UPDATE PHASE
   for each M_j in M
      do repeat obtain N^* from embedding algorithm
         until $S(N^*) = 1$
         $R'(N^*) \leftarrow G(M_j, R(N^*))$
         perform analysis on $R(N^*)$ to obtain $\mathbf{T}(N^*)$
         $N_p \leftarrow \text{parent}(N^*)$
         while N_p is not root
            $\mathbf{T}(N_p) \leftarrow \Sigma_{\mathbf{v} \in \text{children}(N_p)} \mathbf{T}(\mathbf{v})$
            $d(R(N_p)) \leftarrow D(\mathbf{T}(N_p))$
            if $d(G(b, R(N_p))) > \tau$
            then $S(N_p) \leftarrow 0$
               for each N_j in the subtree with root N_p
                  do $S(N_j) \leftarrow 0$
            $N_p \leftarrow \text{parent}(N_p)$

In addition to the embedding protocol described above we also need to specify an extraction protocol. The extraction has to be modified to handle identification of the regions that were avoided during embedding. This can be done in a number of ways, of which we discuss two. One is by providing the extraction algorithm with the fixed threshold that was used to identify these avoided regions. This threshold information should be secret and known only to the extractor and the embedder. It may as well be embedded in the stego object in a way that the extractor can recover it before starting to extract $\mathcal{M}$. This has a couple of drawbacks. First, it imposes a constraint on embedding, namely, that the modifications done for the purpose of embedding do not cause an increase above that threshold. Second, as pointed out to us by an anonymous reviewer, it makes possible a "try-all-thresholds" attack whereby the attacker exploits the fact that there exists a threshold below which nothing was avoided at embedding time. These problems are mitigated by the fact that even though the attacker can successfully find the fixed threshold and restrict the region of attack to a smaller area, it will be harder to apply statistical attacks on that area since this region was picked for embedding for the reason that it was considered to be less vulnerable to statistical attacks.

An alternative mechanism to identify the avoided regions, one that avoids both drawbacks (but that sacrifices some capacity), would consist of augmenting the original message $\mathcal{M}$ with *markers* that identify the avoided regions. One way to do this is by embedding information about each forbidden region immediately prior (or after) that region – e.g., through a special marker symbol followed by avoided-region size. The tree structure should then be used to keep track of the boundaries of avoided regions in order to decrease the amount of bandwidth used up for such marking. At extraction time, the extractor will use this marker information to ignore the avoided regions. Note that, in this second scheme, we no longer impose the constraint that the embedding does not cause a used region to exceed the threshold
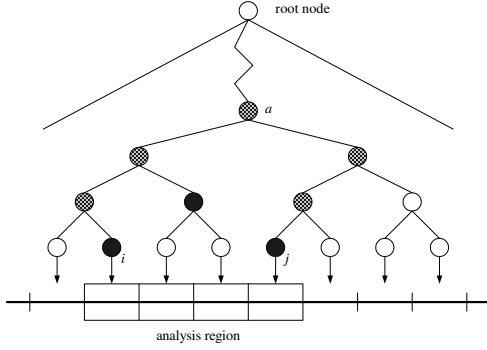
Figure 2: Example of how the hierarchical representation efficiently keeps track of the changes done in the cover document for the one-dimensional case.
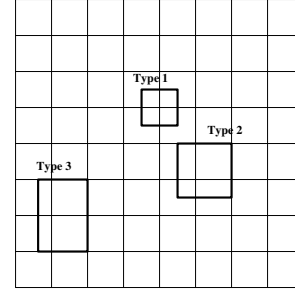

Figure 3: Three basic types of regions at a fixed height h of a quad-tree $\mathcal{T}$ that are used to decompose any arbitrary region at this height.
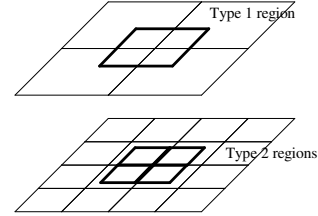

Figure 4: Decomposition of a type R_1 region

τ used to identify avoided regions (although of course we would impose a constraint to not exceed some other threshold $\tau' > \tau$); in this manner there is no threshold below which all was used and none avoided. Having more than one threshold can be achieved by increasing the threshold after the initialization phase. This way if embedding causes a region's statistics to exceed initial threshold, τ, but keeps them below τ', the embedding will still be allowed. If embedding causes a higher increase in statistics that exceeds τ', the algorithm should restore the original values of the region and mark the region as avoided.

3.4   The Upper Bound on detectability

If a message embedding algorithm, is used in conjunction with the proposed protocol to monitor the statistical properties of a cover object, we are able to prove an upper bound on the detectability of the statistical features of a region of arbitrary shape in the stego-object. This upper bound provably provides robustness against attacks based on statistical analysis of the anomalies in a region of the object such as the sliding window in the generalized chi-square attack [11]. The proof we give relies on the fact that any such region can be decomposed into one or more blocks corresponding to the internal and leaf nodes of the tree structure. In the specific case of watermarking, Merhav et.al. [15] have shown that if a maximum distortion constraint can be imposed on the embedding, it is possible to quantify the capacity of the watermarking system in an information theoretic model with a non-malicious adversary.

Using the threshold of the detectability for each node as τ and an additive detectability model, where $d(R(N_i)) = \sum d(R(\text{children}(N_i)))$, we show that for any region $R(N_i)$ in the document the detectability, $d(R(N_i))$, will be

- $O(\tau \log_2 n)$ for one dimensional data with a binary tree representation (e.g., audio, natural language text, software, streaming data)

- $O(\tau \sqrt{n})$ for two-dimensional data with a quad-tree representation(e.g., images).

Suppose that we are interested in obtaining the statistical properties, $\mathbf{T}(R)$, of an arbitrary region R of the one dimensional cover object shown in Figure 2. The region R is bounded by the elementary blocks $R(N_i)$ and $R(N_j)$. The smallest set of nodes selected to represent R are called *representative nodes* and are shown in black in the figure. $\mathbf{T}(R)$ may then be obtained using only these representative nodes. The number of these nodes can be shown to be $O(\log_2 n)$ using the following argument: First, we search for nodes N_i and N_j starting from the root node. Let N_a be the common ancestor of nodes N_i and N_j with smallest height. We find the paths from N_a to the node N_i and pick all the right children of the nodes on the path and similarly pick the left children while searching for N_j from N_a as representative nodes. The shaded nodes in the figure are the nodes visited during this search. By this argument, since the length of the paths from N_a to N_i and N_j will be at most $\log_2 n$ the number of representative nodes will also be $O(\log_2 n)$. If we then sum up the detectability values for these nodes, we get a worst case upper bound of $O(\tau \log_2 n)$ on $d(R)$.

A similar approach can be used to derive an upper-bound in the quad-tree case. We define three basic types of regions, R_1, R_2, and R_3. We use the notation $R_1(h)$ to refer to a type R_1 region at height h. An $R_1(h)$ region does not cover any block in full at height h. An $R_2(h)$ fully covers a block in one corner and partially covers three neighboring blocks at height h. An $R_3(h)$ totally covers two blocks at one side, and partially covers two neighboring blocks height h. Any arbitrary region at height h may be decomposed into a combination of $R_1(h)$, $R_2(h)$, and $R_3(h)$. Refer to Figure 3 which illustrates these regions.

The detectability for $R_1(h)$ is given by $d(R_1(h))$, which we will refer to simply as $d_1(h)$. Similar definitions apply for regions of types R_2 and R_3. We can write the detectability values for regions at height h in terms of detectability values

for regions at lower levels of the tree as

$$d_1(h) \leq 4d_2(h-1) \quad (1)$$
$$d_2(h) \leq \tau + d_2(h-1) + 2d_3(h-1) \quad (2)$$
$$d_3(h) \leq 2\tau + 2d_3(h-1) \quad (3)$$

By using the recursion on $d_3(h)$, we obtain

$$d_3(h) \leq \tau(2^h 3 - 2) \quad (4)$$
$$= O(\tau 2^h) \quad (5)$$
$$= O(\tau\sqrt{n}) \quad (6)$$

where we have used the fact that $h = \log_4 n$. We can use this result to solve for $f_2(h)$ as

$$d_2(h) = \tau + d_2(h-1) + 2O(\tau\sqrt{n}) \quad (7)$$
$$= \tau \log_4 n + \tau + 2O(\tau\sqrt{n}) \quad (8)$$
$$= O(\tau\sqrt{n}) \quad (9)$$

which shows that $f_1(h) = O(\tau\sqrt{n})$.

4. EXPERIMENTAL RESULTS

We have performed experiments to illustrate the effectiveness of our protocol in increasing the stealthiness of a steganographic algorithm. For our experiments we have chosen a simple least significant bit (LSB) embedding steganographic algorithm for color TIFF images, however any other embedding scheme may be employed. A quad-tree structure is used for $\mathcal{T}$.

The embedding algorithm first pads $\mathcal{M}$ with random bits to produce a message $\mathcal{M}'$ with a size in bits equal to the number of pixels in $\mathcal{C}$. $\mathcal{M}$ is located at a random place within $\mathcal{M}'$. A small part of $\mathcal{M}'$ is used to for storing the starting point of $\mathcal{M}$ within $\mathcal{M}'$ and the size of $\mathcal{M}$. Both red and green planes of $\mathcal{C}$ are used for embedding. Each pixel of $\mathcal{C}$ carries only one bit of $\mathcal{M}'$. Bits of $\mathcal{M}'$ are XOR'ed with a random bit, which is generated by a pseudo random bit generator that takes the stego key as a seed. This randomizes the bits of $\mathcal{M}'$. The embedding length is equal to the number of pixels in $\mathcal{C}$. The message length, length of $\mathcal{M}$, is smaller than the number of pixels in $\mathcal{C}$.

The elementary blocks in $\mathcal{C}$ were chosen to be 8×8 pixel blocks. For the experiments reported in this paper we chose the pixel variance of the elementary blocks as the statistical information at the leaf nodes, or $T(N_i) = \text{Var}(R(N_i))$. For internal nodes, we have $\sum_{\mathbf{v} \in \text{children}(N_i)} \mathbf{T}(\mathbf{v})$. The detectability measure for N_i was simply selected to be equal to $-T(N_i)$, in other words, we have

$$d(R(N_i)) = \begin{cases} -\text{Var}(R(N_i)), & \text{for leaf nodes} \\ \sum d(R(\text{children}(N_i))), & \text{for internal nodes} \end{cases}$$

This choice is motivated by the following observation. Usually the message $\mathcal{M}$ that is embedded is an encrypted version of the secret message to be sent, in which case, the sequence of bits in $\mathcal{M}$ will have noise-like characteristics, which will cause an increase in the variance of $\mathcal{C}$. Let the variance of a region of the cover image be σ_c^2 and suppose that after message embedding the variance of that region increases to $\sigma_s^2 = \sigma_c^2 + \epsilon$. For regions with small σ_c^2, the contribution ϵ may make the region visible to steganalysis. Therefore, regions with high variance should have low detectability values and are suitable for embedding. A sample

Figure 5: A sample cover image.

image and the corresponding 8×8 block variances are shown in Figure 5 and Figure 6, respectively.

A quad-tree structure $\mathcal{T}$ is initialized using the initialization phase of the algorithm given in Section 3.3. Let V_h be the set of nodes at height h of $\mathcal{T}$. For each height, h, we calculate the threshold on detectability values, τ_h, as

$$\tau_h = c \left(\frac{1}{|V_h|} \sum_{N_i \in V_h} d(R(N_i)) - \min_{N_i \in V_h} d(R(N_i)) \right) \quad (10)$$

where c is a parameter that controls the number of suitable regions selected. In our experiments we have chosen $c = 0.5$.

The suitability of the node N_i is set using

$$S(N_i) = \begin{cases} 1 & \text{if } d(R(N_i)) < \tau_{h(N_i)} \\ & \quad d(R(\text{parent}(N_i))) < \tau_{h(N_i)+1} \\ 0 & \text{otherwise} \end{cases}$$

Note that the detectability values of both N_i and $\text{parent}(N_i)$ are taken into consideration in deciding if $R(N_i)$ is a suitable region. This is a relaxation on the algorithm described in Section 3.3 in order to avoid setting large blocks of $\mathcal{C}$ as unsuitable for embedding and also taking into account the detectability measures of the siblings of N_i, which are reflected in $d(\text{parent}(N_i))$. This relaxation can be tuned to take into account ancestors of N_i that are further up in $\mathcal{T}$ than $\text{parent}(N_i)$ for achieving better stealthiness.

During the embedding, our protocol restricts the embedding system to use only the suitable regions. The unsuitable regions after the initialization phase of the algorithm for the image in Figure 5 are shown in white in Figure 7. After the final phase of the algorithm the number of unsuitable regions increase for this image, as you can see in Figure 8.

Figure 11 and Figure 12, show the difference images between the cover image shown in Figure 5 and stego-images produced using two different approaches. The gray regions in Figure 12 represent the regions that are the same in both the cover and stego images. From these images it can be seen that our protocol guided the embedding algorithm to avoid regions with high variance.

We tested the performance of our system using the steganalysis attack proposed in [16]. Since the feature extraction of this system was designed for grayscale images, we processed the red, green and blue channels independently. In our experiments we used 141 TIFF images of size 512×512

Figure 6: Variances of elementary blocks of the sample image. Higher values are represented by lighter regions. Note that variance values are inversely proportional to detectability.

Figure 8: Final suitability map for sample image. The regions shown in white are the ones that are labeled as unsuitable for embedding.

classification method	plain embedding	embedding with hierarchical protocol
SVM	%49.65	%42.65
FLD	%76.92	%69.23

Table 1: Classification results.

Figure 7: Initial suitability map for sample image. The regions shown in white are the ones that are labeled as unsuitable for embedding.

pixels obtained from the Watermark Evaluation Testbed (WET) [14].

In order to perform the classification between cover and stego images we have used both support vector machine (SVM) and the Fisher linear discriminant (FLD) classifiers. LIBSVM tools [4] were used for SVM classification. Given the embedding algorithm itself randomizes the message, we inserted a text message, the first chapter of the Tale of Two Cities by Charles Dickens [12]. Although, actual message length is 18%, embedding length is 100% for plain embedding, and it varies for each image when embedding is combined with the protocol. While we force the system to stay out of avoided regions, we decrease the size of random part of $\mathcal{M}'$. The average embedding length was 42% for the embedding with the protocol.

The accuracy of classification for the images in our test set are given in Table 1. Although both classifiers are not very accurate at detecting LSB embedding, from this table it can be seen that our protocol was still able to decrease the detectability of the steganographic method.

We have also performed tests using *RS steganalysis* [9] over the green and red color planes which were used as the embedding channel. Our aim was not to evaluate RS steganalysis *per se* but rather to evaluate the impact of our technique on increasing the stealthiness against statistical steganalysis. This attack is specifically designed to detect LSB embedding. However, as it is also stated in [9] and [8], RS steganalysis is more successful with grayscale images and for messages that are randomly scattered over the stego-image. This is not the case for our embedding algorithm. Even with the plain embedding the error rates were high, because the LSB algorithm perturbs LSBs of all pixels. Therefore, estimated embedding lengths are sometimes
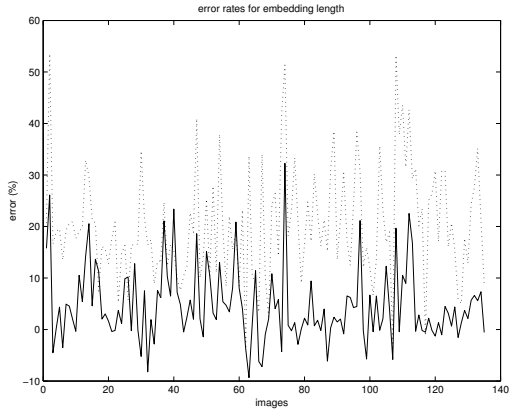
Figure 9: Error of RS-Analysis for the green channel using LSB embedding only and using LSB embedding with hierarchical protocol
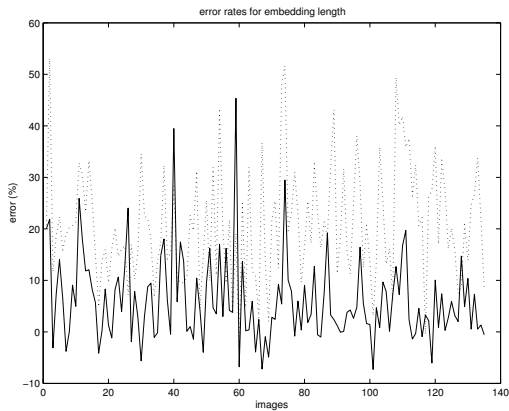
Figure 10: Error of RS-Analysis for the red channel using LSB embedding only and using LSB embedding with hierarchical protocol

higher than 100%. Still, detection errors increase when our protocol is used, as you can see in Figure 9 for green color plane and in Figure 10 for red color plane.

5. CONCLUSIONS

We described, implemented, and tested a protocol for improving the stealthiness of information-hiding schemes. Although our protocol does not completely eliminate the statistical anomalies caused by embedding that are a major threat to the embedding algorithm's stealthiness, it effectively controls their severity and decreases their total number.

Guided by a continuously updated detectability representation of the cover object, our protocol provides a mechanism for controlling statistical anomalies at both fine and coarse scales of granularity. We use a hierarchical representation to manage the complexity of dynamically keeping track of the detectability of the cover object during embedding.

We also quantify how bounds on the detectability of regions from the hierarchy translate into detectability bounds for arbitrary regions.
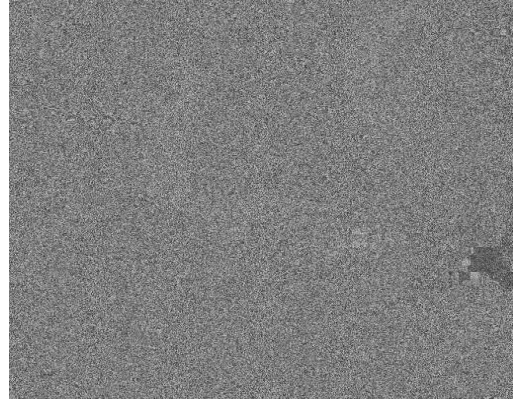
Figure 11: Difference of cover image and stego image generated using LSB embedding only
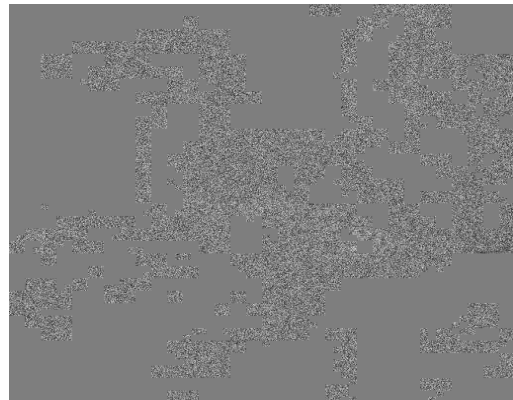
Figure 12: Difference of cover image and stego image generated using LSB embedding with hierarchical protocol

6. REFERENCES

[1] R. J. Anderson and F. A. Petitcolas. On the Limits of Steganography. *IEEE Journal of Selected Areas in Communications*, 16(4):474 – 481, May 1998.

[2] M. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg. Natural Language Watermarking and Tamperproofing. In *Fifth Information Hiding Workshop*, volume LNCS, 2578, Noordwijkerhout, The Netherlands, October, 2002. Springer-Verlag.

[3] J. T. Brassil, S. Low, and N. F. Maxemchuk. Copyright protection for electronic distribution of text documents. *Proceedings of the IEEE (USA)*, 87(7):1181–1196, 1999.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] H. Farid. Detecting Steganographic Message in Digital Images. Technical Report TR2001-412, Darmouth College, Computer Science, Hanover, NH, USA, 2001.

[6] E. Franz. Steganography Preserving Statistical Properties. In *Fifth Information Hiding Workshop*, volume LNCS, 2578, Noordwijkerhout, The Netherlands, October 2002. Springer-Verlag.

[7] J. Fridrich and M. Goljan. Digital image staganography using stochastic modulation. In *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents*, volume 5020, pages 191–202, San Jose, CA, 21 – 24 January 2003.

[8] J. Fridrich and M. Goljan. Practical Steganalysis of Digital Images - State of the Art. In *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents*, volume 4675, pages 1–13, San Jose, CA, USA, January, 2002.

[9] J. Fridrich, M. Goljan, and R. Du. Reliable Detection of LSB Steganography in Color and Grayscale Images. In *Proceedings of the ACM Workshop on Multimedia and Security*, pages 27–30, Ottowa, Canada, 5 October 2001.

[10] J. Fridrich, M. Goljan, and D. Hogea. New Methodology for Breaking Steganographic Techniques for JPEGs. In *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents*, volume 5020, pages 143–155, San Jose, CA, 21 – 24 January 2003.

[11] J. Fridrich, M. Goljan, and D. Soukal. Higher-Order Statistical Steganalysis of Palette. In *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents*, volume 5020, pages 178–190, San Jose, CA, 21 – 24 January 2003.

[12] M. Hart. *Project Gutenberg*. http://www.gutenberg.net/, 2004.

[13] S. Katzenbeisser and F. Petitcolas(Ed.). *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, 2000.

[14] H. C. Kim, H. Ogunley, O. Guitart, and E. J. Delp. The watermark evaluation testbed. In *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents*, San Jose, CA, 18 – 22 January 2004.

[15] A. Levy and N. Merhav. An image watermarking scheme based on information theoretic principles. Technical Report HPL-2001-13, HPL Technical Report, January 2001.

[16] S. Lyu and H. Farid. Detecting Hidden Messages using Higher-Order Statistics and Support Vector Machines. In *Proceedings of the Fifth Information Hiding Workshop*, volume LNCS, 2578, Noordwijkerhout, The Netherlands, October, 2002. Springer-Verlag.

[17] A. Pfitzmann and A. Westfeld. Attacks on steganographic systems. In *Third Information Hiding Workshop*, volume LNCS, 1768, pages 61–76, Dresden, Germany, 1999. Springer-Verlag.

[18] N. Provos. Defending Against Statistical Steganalysis. In *Proceedings of 10th USENIX Security Symposium*, Washington DC, USA, August 2001.

[19] P. Sallee. Model-based steganography. In *International Workshop on Digital Watermarking*, Seoul, Korea, 20–22 October 2003.

[20] A. Westfeld. F5-A Steganographic Algorithm: High Capacity Despite Better Steganalysis. In *Fourth Information Hiding Workshop*, volume LNCS, 2137, pages 289–302, Pittsburgh, USA, April 2001. Springer-Verlag.

[21] J. Zollner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotrascke, A. Westfeld, G. Wicke, and G. Wolf. Modeling the Security of Steganographic Systems. In *Second Information Hiding Workshop*, volume LNCS, 1525, Portland, Oregon, USA, 1999. Springer-Verlag.