Blind Steganography Detection Using a Computational Immune System: A Work in Progress

Jacob T. Jackson, Gregg H. Gunsch, Roger L. Claypoole, Jr., Gary B. Lamont†
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology

Abstract

Research in steganalysis is motivated by the concern that communications associated with illicit activity could be hidden in seemingly innocent electronic transactions. By developing defensive tools before steganographic communication grows, computer security professionals will be better prepared for the threat. This paper proposes a computational immune system (CIS) approach to blind steganography detection.

1 Introduction

Most current steganalytic techniques are similar to virus detection techniques in that they tend to be signature-based, and little attention has been given to blind steganography detection using an anomaly-based approach, which attempts to detect departures from normalcy. While signature-based detection is accurate and robust, anomaly-based detection can provide flexibility and a quicker response to novel techniques. Using anomaly-based detection in conjunction with signature-based detection will enhance the layered approach to computer defense.

The research proposed here is incomplete. Much of the background work has been done and the development of the methodology is nearing completion. The chosen problem domain is discussed in Section 2 and the necessary background information is summarized in Section 3. The methodology is presented in Section 4, initial results are given in Section 5, and Section 6 contains a short summary.

2 Problem Description

The goal of digital steganography is to hide an embedded file within a cover file such that the embedded file's existence is concealed. The resulting file is called the stego file. Steganalysis is the counter to steganography and its first goal is detection of steganography.

2.1 Steganography Overview

† The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

There are many approaches to hiding the embedded file. The embedded file bits can be inserted in any order, concentrated in specific areas that might be less detectable, dispersed throughout the cover file, or repeated in many places. Careful selection of the cover file type and composition will contribute to successful embedding.

A technique called *substitution* replaces cover file bits with embedded file bits. Since the replacement of certain bits in the cover file will be more detectable than the replacement of others, a smart decision has to be made as to which bits would make the best candidates for substitution. The number of bits in the cover file that get replaced will also affect the success of this method. In general, with each additional bit that is replaced the odds of detection increases, but in many cases more than one bit per cover file byte can be replaced successfully. Combining the correct selection of bits with analysis of the maximum number of bits to replace should result in the smallest possible impact to the statistical properties of the cover file. (Katzenbeisser, 2000)

One of the more common approaches to substitution is to replace the least significant bits (LSBs) in the cover file (Katzenbeisser, 2000). This approach is justified by the simple observation that changing the LSB results in the smallest change in the value of the byte. One significant advantage of this method is that it is simple to understand and implement and many steganography tools available today use LSB substitution.

The Discrete Cosine Transform (DCT) is the keystone for JPEG compression and it can be exploited for information hiding. For one technique, specific DCT coefficients are used as the basis of the embedded file hiding. The coefficients correspond to locations of equal value in the quantization table. The embedded file bit is encoded in the relative difference between the coefficients. If the relative difference does not match the bit to be embedded, then the coefficients are swapped. This method can be enhanced to avoid detection if blocks that are drastically changed by swapping the coefficients are not used for hiding. A slight variation of this technique is to encode the embedded file in the decision to round the result of the quantization up or down. (Katzenbeisser, 2000) Other steganographic techniques, including spread spectrum, statistical steganography, distortion, and cover generation, are described in detail in (Katzenbeisser, 2000).

2.2 Steganalysis Overview

Though the first goal of steganalysis is detection, there can be additional goals such as disabling, extraction, and confusion. While detection, disabling, and extraction are self-explanatory, confusion involves replacing the intended embedded file (Katzenbeisser, 2000). Detection is more difficult than disabling in most cases, because disabling techniques can be applied to all files regardless of whether or not they are suspected of containing an embedded file. For example, a disabling scheme against LSB substitution in BMP image files would be to use JPEG compression on all available BMP files (Johnson, 2001). However, if only a few files are suspected to have embedded files, then disabling in this manner is not very efficient.

One steganalytic technique is visible detection, which can include human observers detecting minute changes between a cover file and a stego file or it can be automated. For palette-based images if the embedded file was inserted without first ordering the cover file palette according to

color, then dramatic color shifts can be found in the stego file. Additionally, since many steganography tools take advantage of close colors or create their own close color groups, many similar colors in an image palette may make the image become suspect (Johnson, 2001). By filtering images as described by Westfeld and Pfitzmann in (Westfeld, 2000), the presence of an embedded file can become obvious to the human observer.

Steganalysis can also involve the use of statistical techniques. By analyzing changes in an image's close color pairs, the steganalyst can determine if LSB substitution was used. Close color pairs consist of two colors whose binary values differ only in the LSB. The sum of occurrences of each color in a close color pair does not change between the cover file and the stego file (Westfeld, 2000). This fact, along with the observation that LSB substitution merely flips some of the LSBs, causes the number of occurrences of each color in a close color pair in a stego file to approach the average number of occurrences for that pair (Johnson, 2001). Determining that the number of occurrences of each color in a suspect image's close color pairs are very close to one another gives a strong indication that LSB substitution was used to create a stego file (Westfeld, 2000).

Fridrich and others proposed a steganalytic technique called the RQP method. It is used on color images with 24-bit pixel depth where the embedded file is encoded in random LSBs. RQP involves inspecting the ratio between the number of close color pairs and all pairs of colors. This ratio is calculated on the suspect image, a test message is embedded, and the ratio is calculated again. If the initial and final ratios are vastly different then the suspect image was likely clean. If the ratios are very close then the suspect image most likely had a secret message embedded in it. (Fridrich, 2000)

These statistical techniques benefit from the fact that the embedding process alters the original statistics of the cover file and in many cases these first-order statistics will show trends that can raise suspicion of steganography. (Fridrich, 2000, Westfeld, 2000). However, steganography tools such as OutGuess (Provos, 2002) are starting to maintain the first-order statistics during the embedding process. Steganalytic techniques using sensitive higher-order statistics have been developed to counter this covering of tracks (Farid, 2001, Fridrich, 2002).

Farid developed a steganalytic method that uses deviation from expected statistics as an indication of a potential hidden message. The training set for his Fisher linear discriminant (FLD) analysis consisted of a mixture of clean and stego images. He then tested the trained FLD on a previously unseen mixture of clean and stego images. He did this separately for Jpeg-Jsteg (Upham, 2002), EzStego (Machado, 2002) and OutGuess. The features that he was training and testing on were based upon particular statistics gathered from a wavelet decomposition of each image. Farid's work will be discussed in more detail later because it will be heavily leveraged in this research. (Farid, 2001)

2.3 Research Goal and Hypothesis

The goal of this research is **to develop CIS classifiers, which will be evolved using a genetic algorithm (GA), that distinguish between clean and stego images by using statistics**

gathered from a wavelet decomposition. With successful classifiers the foundation for a CIS is established, but the development of a complete CIS is beyond the scope of this research. Additionally, prediction of embedded file size, prediction of the stego tool, and extraction are also beyond the scope of this research and might not even be possible using the proposed techniques.

Our initial research hypothesis is:

> ***CIS classifiers evolved using genetic algorithms will be able to distinguish between clean and stego images with results that are at least as promising as previous similar wavelet decomposition steganalysis research that used pattern recognition.***

The hypothesis alludes to Farid's research (Farid, 2001) and is based on the fact that wavelet decomposition is a common theme. The terms and concepts that are presented in the research goal and hypothesis will be further explained in the following section.

3 Background

3.1 Wavelet Analysis of Images

In signal processing there are numerous examples of the benefits of working in the frequency domain. Fourier analysis remains a powerful technique for transforming signals from the time domain to the frequency domain. However, time information is hidden in the process. In other words, the time of a particular event cannot be discerned from the frequency domain view without performing phase calculations, which is very difficult for practical applications. (Hubbard, 1996)

The Fourier transform was modified to create the Short-Time Fourier Transform (STFT) in an attempt to capture both frequency and time information. The STFT repeatedly applies the Fourier transform to disjoint, discrete portions of the signal of constant size. Since the time window is constant throughout the analysis, a signal can be analyzed with high time precision or frequency precision, but not both (Rioul, 1991). As the window gets smaller, high frequency, transitory events can be located, but low frequency events are not well represented. Similarly as the window gets larger, low frequency events are well represented, but the location in time of the interesting, high frequency events becomes less precise. (Hubbard, 1996)

Wavelet analysis offers more flexibility because it provides long time windows for low frequency analysis and short time windows for high frequency analysis as is shown in Figure 1. As a result, wavelet analysis can better capture the interesting transitory characteristics of a signal. (Rioul, 1991)

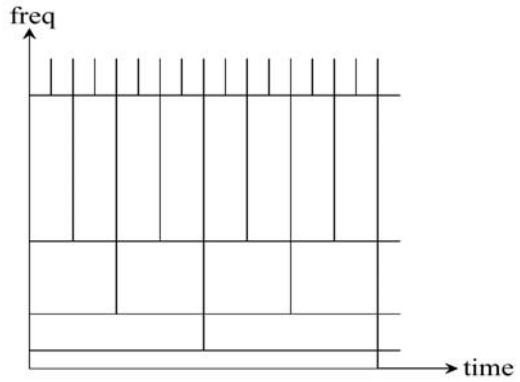

Figure 1. Wavelet Analysis

Figure 2. Daubechies 8 Wavelet

A wavelet is a waveform of limited duration with an average value of zero. Figure 2 shows an example of a wavelet. One-dimensional wavelet analysis decomposes a signal into basis functions that are shifted and scaled versions of a *mother* wavelet. Wavelet coefficients are generated and are a measure of the similarity between the basis function and signal being analyzed. (Rioul, 1991)

To scale a wavelet is to compress or extend it along the time axis. A compressed wavelet will produce higher wavelet coefficients when evaluated against high frequency portions of the signal. Therefore, compressed wavelets are said to capture the high frequency events in a signal. A smaller scale factor results in a compressed wavelet because scale and frequency are inversely proportional. (Math Works, 2001)

An extended wavelet will produce higher wavelet coefficients when evaluated against low frequency portions of the signal. As a result, extended wavelets capture low frequency events and have a larger scale factor (Math Works, 2001). Scale offers an alternative to frequency and leads to a *time-scale* representation that is convenient in many applications (Rioul, 1991).

Though the above discussion of Fourier analysis and wavelet analysis made reference to the time and frequency domains typically associated with signal processing, the concepts also apply to the spatial and spatial frequency domains associated with image processing.

There are different types of wavelet transforms, including the Continuous Wavelet Transform (CWT) and the Discrete Wavelet Transform (DWT). The CWT is used for signals that are continuous in time and the DWT is used when a signal is being sampled, such as during digital signal processing or digital image processing.

The DWT has a scaling function ø and a wavelet function φ associated with it. The scaling function can be implemented using a low pass filter and is used to create the scaling coefficients that represent the signal approximation. The wavelet function can be implemented as a high pass filter and is used to create the wavelet coefficients that represent the signal details. If the DWT is used by scaling and shifting by powers of two (dyadic), the signal will be well represented and the decomposition will be efficient and easy to compute. In order to apply the DWT to images, combinations of the filters (combinations of the scaling function and the wavelet function) are used first along the rows and then along the columns to produce unique subbands. (Rioul, 1991)

The LL subband is produced by low pass filtering along the rows and columns and is commonly referred to as a course approximation of the image because the edges tend to smooth out. The LH subband is produced by low pass filtering along the rows and high pass filtering along the columns, thus capturing the horizontal edges. The HL subband is produced by high pass filtering along the rows and low pass filtering along the columns, thus capturing the vertical edges. The HH subband is produced by high pass filtering along the rows and columns, thus capturing the diagonal edges. The LH and HL subbands are considered the bandpass subbands and the LH, HL, and HH subbands together are called the detail subbands. These subbands are shown in Figure 3. (Mendenhall, 2001) By repeating the process on the LL subband, additional scales are produced. In this context scales are synonymous to the detail subbands.

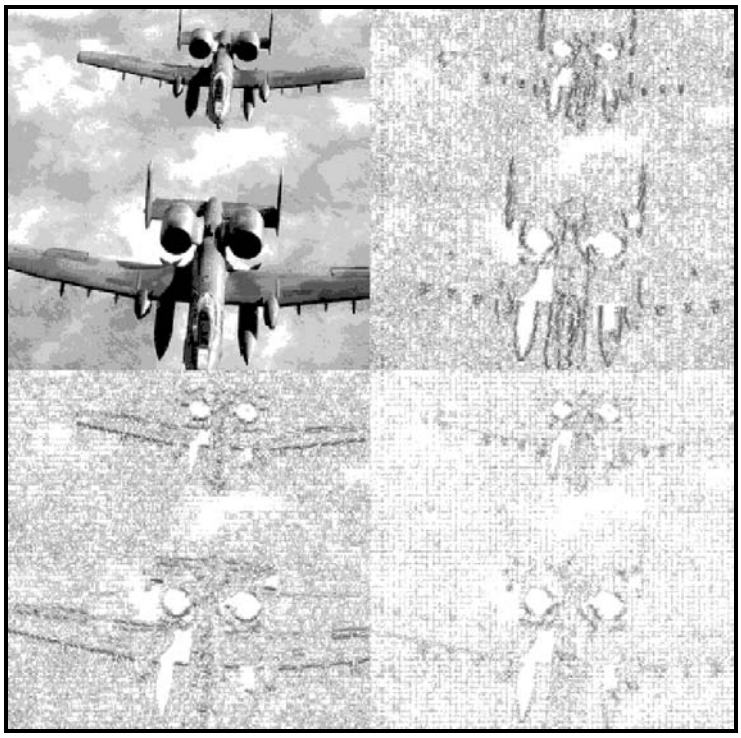

Figure 3. Wavelet decomposition using Daubechies (7,9) biorthogonal filters. LL subband on upper left, LH on lower left, HL on upper right, and HH on lower right. The LH, HL, and HH subbands have been inverted and rescaled for ease of viewing.

The statistics of the generated coefficients of the various subbands offer valuable results. According to Farid, a broad range of natural images tends to produce similar coefficient statistics. Additionally, alterations such as steganography tend to change those coefficient statistics. The alteration was enough to provide a key for steganography detection in Farid's research. (Farid, 2001)

One set of statistics that Farid used consisted of the mean, variance, skewness, and kurtosis of the coefficients generated at the LH, HL, and HH subbands for all scales. If s is the number of scales represented in a decomposition then the number of individual statistics collected on the actual coefficients is $12(s - 1)$. He also gathered statistics from an optimal linear predictor of coefficient magnitude, which was implemented using linear regression. It used nearby coefficients and coefficients from other subbands and other scales to predict the value of a particular coefficient such that the error between the predicted value and the observed value was minimized. Farid's choice of predictor coefficients was based upon similar work presented in (Buccigrossi, 1999). Statistics were gathered on the resulting minimized errors and included the mean, variance, skewness, and kurtosis. This also resulted in $12(s - 1)$ individual statistics for a total of $24(s - 1)$. Since s was four in Farid's research, 72 individual statistics were generated. (Farid, 2001)

Farid was able to predict coefficients because of the clustering and persistence properties of the DWT. Clustering means that wavelet coefficients tend to group together according to

magnitude. In other words, adjacent coefficients tend to have similar magnitudes. Persistence means that large and small coefficients tend to be represented the same in different scales. This can be seen by observing a multi-scale wavelet decomposition of an image such as that in Figure 4. Different scales display a similar representation of the image at different resolutions. (Mendenhall, 2001)

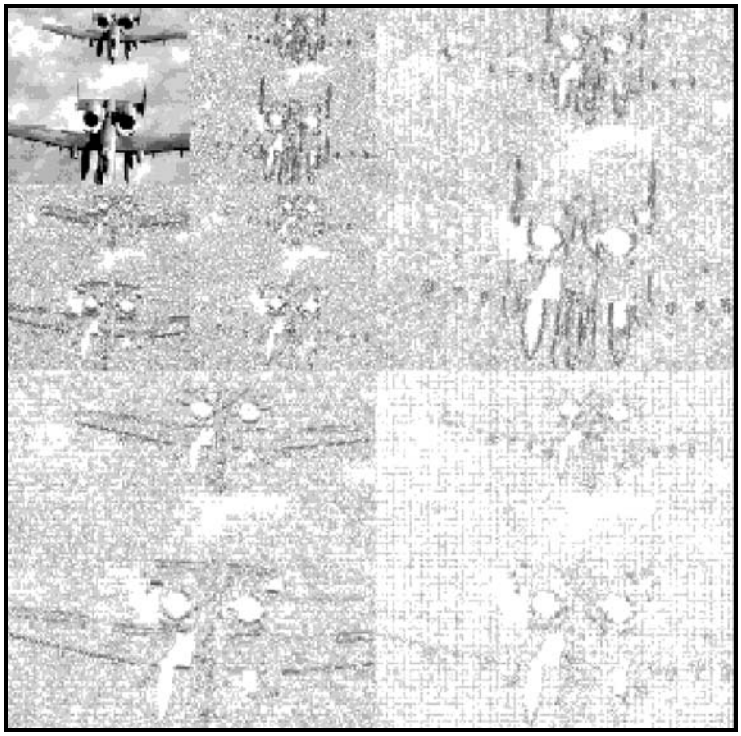

Figure 4. Two iterations of wavelet decomposition using Daubechies (7,9) biorthogonal filters showing clustering and persistence. The LH, HL, and HH subbands at each scale have been inverted and rescaled for ease of viewing.

Farid's results were highly dependent on the particular steganographic method. He achieved detection rates ranging from 97.8% with 1.8% false positives for Jpeg-Jsteg to 77.7% with 23.8% false positive rate for OutGuess with statistical correction. Accepting a smaller detection rate (small detection rate drop for Jpeg-Jsteg and a large drop for OutGuess with statistical correction) can lower the false positive rate. Since the steganography programs chosen for Farid's analysis most likely represent the range of detection ease (Jpeg-Jsteg – easy detection, OutGuess - difficult detection), he concluded that his method would be just as successful on other known methods. Also, the ratio of embedded file size to cover file size will typically affect the accuracy of just about any steganalytic technique and this method is no exception. (Farid, 2001)

3.2 Computational Immune Systems (CIS)

A CIS attempts to closely model particular features of the biological immune system (BIS) that could present a solution to a computational problem. Major BIS elements of interest include

multi-layered protection, highly distributed detection and memory systems, diversity of detection ability across individuals, inexact matching strategies, and sensitivity to most new foreign patterns (Forrest, 1996). The major problem for both biological and computational immune systems is to distinguish between *self* and *nonself*. The immunology problem is further complicated by the fact that the definitions of self and nonself shift over time. In the computational environment self can be thought of as allowable activity and nonself can be thought of as prohibited or anomalous activity.

Possible approaches to distinguishing between self and nonself include the use of pattern recognition or neural networks. Another approach deploys a structure within a CIS that interacts with suspect data in order to determine if the data is self or nonself. This structure can be called a *classifier*, *antibody* (Williams, 2001), or *detector* (Anchor, 2002). For this research the term classifier will be used.

3.2.1 Classifier Creation and Negative Selection

An initial population of potential classifiers must be established and this is typically done in a random fashion so that the solution space is well covered. Negative selection eliminates classifiers that match self and is usually done in conjunction with the initial generation. (Williams, 2001)

3.2.2 Affinity Maturation Using GAs

The random classifiers will inevitably have room for improvement. The use of GAs to improve classifiers has been shown to be a viable approach in other problem domains (Williams, 2001).

"A GA performs a multi-directional search" for the best solution to a computational problem "by maintaining a population of potential solutions and encourages information formation and exchange between these directions" (Michalewicz, 1996). During iteration t of the genetic algorithm the population of possible solutions undergoes an evaluation test in the form of a fitness function. During iteration $t + 1$, a portion of the survivors from iteration t are altered using crossover and mutation and then processed again with the evaluation function. Crossover is achieved by swapping solution features to create next generation solutions that have exchanged pieces of information. Mutation alters a small piece of a solution in order to introduce extra variability into the population. (Michalewicz, 1996)

The terms *gene* and *chromosome* are used in the GA context. Chromosomes represent solutions to the particular problem and consist of genes, which represent the features of a particular solution.

Crossover

Crossover typically results in a rapid exploration of the solution space. For solutions that are represented by bit strings, crossover is accomplished by dividing the solution into two or more

disjoint segments and interchanging segments between solutions. Not all solutions have to be selected for crossover and the typical crossover probability is between 0.6 and 1. (Beasley, 1993a)

Single point crossover is traditionally used and involves dividing the solution into two segments. However, two-point crossover can also be used. Two-point crossover is accomplished by dividing the solution into three segments and exchanging one of them with another solution. The number of segments is not limited to three when selecting crossover points. Uniform crossover is another technique that involves merging two parent solutions into an offspring solution based upon a mask. If the mask is viewed as a bit string, then the parent that donates a particular bit to the offspring is determined by the bits in the mask. (Beasley, 1993b)

Mutation

Mutation provides a mechanism for bringing diversity to a population (Luke, 1997) and usually results in a slow random search of the solution space (Beasley, 1993a). It is beneficial because it helps ensure that no solution has zero probability of being examined. For solutions that are represented by bit strings, mutation is typically accomplished by flipping bits in the solutions with a small probability - typically between 0.001 and 0.01. (Beasley, 1993a)

Convergence

Convergence occurs in single objective problems when the fitness of the population becomes uniform around the best solution after a number of generations. This can be quantitatively defined for genes as the point when identical genes occur in 95% of the population. All of the genes have to converge before the population is said to have converged. (Beasley, 1993a)

Fitness Function

GA parameters such as crossover and mutation probability and population size can be selected from a large range and the GA can still be successful as long as the fitness function is accurate. The ideal fitness function should be smooth and regular with solutions of similar fitness lying close together. Since the ideal fitness function is likely nonexistent, practical fitness functions should have few local maxima or an obvious global maximum. (Beasley, 1993a)

Some problems that stem from fitness functions include premature convergence and slow finishing. Premature convergence occurs when solutions with high fitness function results begin to dominate the population very early. Premature convergence could mean that a local maximum has been found due to lack of exploration and it can be countered by not working with raw fitness scores and ensuring that the population remains somewhat diverse. Slow finishing results when the population is mostly converged and is having a hard time locating the actual maximum. This can be remedied by setting a reasonable limitation on the number of generations that are allowed. (Beasley, 1993a)

Natural Selection

Natural selection determines which solutions should be carried over into the next generation, and there are many ways to achieve this computationally. The particular problem domain determines useful natural selection methods.

Fitness scaling and fitness windowing remap raw fitness scores to prevent premature convergence. Effectively, they readjust the number of opportunities that solutions will have to reproduce. These techniques can lead to problems if a single solution appears that is either drastically more fit or drastically more unfit than all of the other solutions. (Beasley, 1993a)

Fitness ranking eliminates the negative effects that extreme solutions have on remapping. The solutions are ordered according to raw fitness and then a new fitness is assigned to each solution according to where they fall in the order. The new fitness can be scaled in many ways, but linear or exponential scalings are commonly used. There is empirical evidence that fitness ranking works better than both fitness scaling and fitness windowing. (Beasley, 1993a)

Tournament selection randomly selects *n* solutions from the population and compares their fitness scores. The number of competing solutions in the tournament can vary upwards from two with larger tournaments effectively reducing the chances of below average solutions winning tournaments. The opposite can be achieved when there are two tournament competitors (binary tournament selection) by allowing the solution with the highest fitness to win with a probability between 0.5 and 1. Tournament selection can be done with or without replacement of the tournament competitors back into the population. The appropriate choice is determined by a particular application. (Beasley, 1993a)

Steady-state selection typically only replaces two solutions in the population as opposed to replacing the entire population with each generation. Two solutions are selected as parents and two solutions are selected to be replaced with the offspring by some variant of a fitness function. The chosen parents then possibly undergo crossover and mutation to create offspring to fill the empty solution slots. (Beasley, 1993b)

Evolutionary Multiobjective Optimization

A multiobjective approach attempts to present solutions to a problem with several requirements or objectives. One approach uses the Pareto front, a collection of solutions that have no superior in all objectives. The solutions along the Pareto front are also referred to as non-dominated solutions. If a single solution is required, it is selected from those solutions along the Pareto front. (Coello, 1999)

Pareto-based approaches drive the population towards the Pareto front by giving locally non-dominated solutions a better chance to reproduce. Fitness is typically determined by assigning the non-dominated solutions the best fitness score and removing them from further fitness score assignment. Then the non-dominated solutions from the remaining solutions are given the next best fitness score and so on (Coello, 1999). An advantage to this approach is that improvement of a requirement is rewarded regardless of the other requirements. The result is that solutions

that perform well on most requirements will survive natural selection. Regardless of the method used, as the number of requirements and the complexity of those requirements increases, solving the problem becomes more difficult. (Fonesca, 1995)

Another multiobjective approach uses aggregating functions. Aggregating functions combine the objectives in some manner to produce a single fitness function. One method of doing this is to assign weights to a solution's fitness score for each objective and then summing the weighted objective scores into a single fitness score. The main problem with this approach is that it requires careful assignment of the weights so that one objective does not dominate the others unless it should. However, aggregating functions typically are computationally efficient compared to other multiobjective approaches. (Coello, 1999)

3.3 Inexact Matching Function

Once classifiers have been satisfactorily evolved they are deployed against suspect data. If they are required to completely match suspect data before triggering an alarm, then they will not be useful for novel attacks. Additionally, they must be somewhat general in order to be efficient. In general, an inexact match occurs when a subset of classifier features match the equivalent features of the suspect data. The number of features in the subset is determined by the application. (Williams, 2001)

4 Methodology

The general process for this research will include the following:

1. Creation of clean and stego image databases.
2. Wavelet analysis of clean and stego images to generate wavelet coefficients.
3. Gathering of statistics on wavelet coefficients.
4. Evolution of classifiers based upon a subset of the clean wavelet coefficient statistics.
5. Testing of classifiers against clean and stego images.

4.1 Image Formats and Stego Programs

This research will test 8-bit .bmp, .jpg, and .gif image files because they are very common digital image formats. Both grayscale and color .gif images will be tested for reasons to be discussed later. These choices allow for coverage of RGB images (.bmp and .jpg images), indexed RGB images (color .gif images), and indexed images with linear, monotonic colormaps (grayscale .gif images).

These choices also allow for testing of both substitution (.bmp and .gif) and transform steganography techniques (.jpg). There are many steganography tools available that can be used for this selection of formats and techniques, but only EzStego (.bmp and .gif), Jpeg-Jsteg (.jpg), and OutGuess (.jpg) will be tested. This choice is mainly influenced by Farid's research, but it turns out to be a good choice of tools for other reasons. The programs are user-friendly, provide good functionality, and represent a range of detection ease (Jpeg-Jsteg - easy detection, OutGuess with statistical correction - hard detection). (Farid, 2001)The choice of image formats

is not limited by the choice of MATLAB as the tool to perform the wavelet decomposition despite the fact that the MATLAB Wavelet Toolbox can only perform wavelet decompositions on indexed images with linear, monotonic colormaps (i.e., grayscale). The particular implementation of the Daubechies (7,9) biorthogonal filters will also perform the wavelet decomposition, but it also has the same input image requirements. Therefore, both clean and stego images will have to be converted to grayscale in order to accomplish the wavelet decomposition. Farid achieved respectable results despite having to convert his images to grayscale prior to wavelet analysis, which shows that the conversion to grayscale did not remove the all of the effects of the steganography. However, in order to determine the effects of grayscale conversion on the stego image detection ease, grayscale images will be included in the test images in this research.

Several image databases will have to be created in order to accomplish this research. The clean images will be originals taken with a digital camera to avoid the slight possibility that images downloaded from the Internet are already stego images. This is required for proper experimental control because research has shown that embedding in a stego image has quite a different result than embedding in a clean image (Fridrich, 2000). A set of images will be chosen and stored in .bmp, .jpg, color .gif, and grayscale .gif formats to form four clean image databases. Each image of a particular format will be cropped to the same size so that the ratio of embedded file size to cover file size can be consistent and to meet the requirements of the wavelet decomposition.

The maximum embedded file size associated with each cover image format, size, and the appropriate steganography program will be determined. Then embedded image files of various sizes will be created from random croppings of the images in the image database. This choice of embedded files is motivated by Farid, should be easy to implement, and will bring randomness to the embedding. Embedded files that are 1%, 5%, 10%, 25%, 50%, and 100% of the maximum embedded file size will be created in order to vary the ease of detection. Thus there will be six embedded image databases for each of the four clean image databases resulting in a total of 24 embedded image databases. An additional embedded image database will have to be created that is associated with .jpg images if Jpeg-Jsteg and OutGuess do not have the same maximum embedded file size.

A random embedded file of a particular size will be selected and embedded in a clean image using the appropriate selection of EzStego, Jpeg-Jsteg, or OutGuess to create a file for the stego image databases. This will result in 30 stego image databases, one for each embedded file size in the .bmp, .gif, and color .gif formats, and two (due to the use of both Jpeg-Jsteg and OutGuess for .jpg embedding) for each embedded file size for the .jpg format. The goal is for this to be a scripted computational process and one concern with this approach is that some stego programs require a password, thus complicating the embedding process. If passwords are needed they will be the same for each image because the password only determines the pseudorandom dispersement of embedded image bits throughout the cover image.

Each RGB image in the clean and stego databases will be converted to grayscale (grayscale = 0.299*R+0.587*G+0.114*B) prior to undergoing wavelet decomposition. The wavelet decomposition will be accomplished using an implementation of the Daubechies(7,9) biorthogonal filters as implemented in MATLAB by Maj Roger Claypoole. This implementation

will be used because it is convenient and the design and implementation of filters to perform the wavelet decomposition is beyond the scope of this research. These particular filters are commonly used throughout the image processing community (Mendenhall, 2001).

The coefficients from each image's wavelet decomposition will be stored in preparation for gathering the chosen statistics. The statistics will then be gathered from the coefficients and grouped together into a single string. The coefficient magnitude prediction will then be performed and the results stored as well. The statistics from the prediction will be gathered and appended to the string of actual coefficient statistics.

A major concern associated with this research is whether the multidimensional space, or hypervolume, represented by the clean image statistics is disjoint from the hypervolume represented by the stego image statistics. This will be checked after all the statistics are gathered and a determination will be made whether to adjust which statistics are gathered and how to predict coefficients. Another option for predicting coefficients is presented in (Buccigrossi, 1999). One major factor in this decision is the fact that the analysis associated with determining the best coefficient predictors is beyond the scope of this research.

4.2 Classifier Generation and Testing

As mentioned in Section 2.3, the main goal of this research is to evolve CIS classifiers and then deploy them against suspect images. Though these classifiers could be part of a CIS, the complete CIS will not be developed. With the appropriate modifications, the classifiers developed here could be used in a CIS similar to the partially implemented Computer Defense Immune System architecture, described in (Harmer, 2002).

Classifiers will be developed and used in this research because the other techniques for distinguishing between self and nonself, pattern recognition and neural networks, have either already been explored or are computationally infeasible. As mentioned previously, pattern recognition was used in Farid's research (Farid, 2001). Neural networks are a powerful technique when the number of input variables is no more than 30. Since the number of input variables is going to start at 72 for this research, the use of neural networks was ruled out. The number of scales in the wavelet analysis can be reduced to reduce the number of neural network inputs, but that might reduce the effectiveness of the wavelet analysis. Another way to reduce the number of inputs is to reduce the number of statistics that are used, but without any solid analysis such action could not be justified.

For this research, self is defined by the hypervolume or collection of hypervolumes represented by the clean image statistics. Nonself is everything that is not part of self. In other words, nonself is everything that is not in the hypervolume represented by the clean image statistics. The distinction between nonself and the hypervolume represented by the stego image statistics should be made because the two are not necessarily the same. However, it is likely that the hypervolume represented by the stego image statistics is a subset of nonself.

An initial population of classifiers will be randomly generated and then subjected to negative selection. Classifiers will detect nonself in order to be consistent with the typical immune

system scenario, and the result of the random generation and negative selection will be a range of values for each statistic that is outside the self hypervolume. In the GA context, these classifiers can be thought of as chromosomes that consist of genes defined by a particular statistic.

Classifiers will then be evolved using crossover and mutation and the result could easily be that the range of values for some statistics is no longer outside the self hypervolume. Single point crossover will be used and initially there will be no requirement for crossover of similar genes (i.e., the gene that is defined by the mean of the coefficients in the LH subband is similar to the gene that is defined by the mean of the coefficients in the HL subband). Mutation will be achieved by flipping bits within the chromosomes.

Each chromosome in the current population will be subjected to the fitness function that rewards growth of a gene's range and penalizes when a gene impinges on the self hypervolume. This approach is motivated by research described in (Williams, 2001). The fitness function will be a multiobjective aggregating function and the chromosome that covers the largest hypervolume without impinging on self will be given the highest fitness score.

After the fitness scores are calculated for each classifier, binary tournament selection with replacement will be used to select the classifiers that will continue on to the next generation. Tournament selection will help to maintain a diverse population of classifiers, which is beneficial because more of the solution hypervolume will be covered.

It is likely that the number of allowed generations will have to be adjusted to maximize convergence but still allow for efficient computation.

The inexact matching function will signal a match if any part of the suspect image statistics match with the classifier's appropriate statistical range. The inexact matching function will primarily be used after a population of classifiers has converged and is being tested against suspect images, but its algorithms can be reused elsewhere. For example the same algorithms can be used during negative selection to ensure that the initial population of classifiers lies outside the self hypervolume. They could also be used as part of the fitness function to determine whether a penalty should be assessed.

4.3 Research Concerns

As previously mentioned, a major concern is that the clean image statistics and the stego image statistics might not be disjoint. The amount of overlap will determine the false positives and false negatives that are experienced. Anomaly-based detection is plagued by false positives and it often takes considerable effort to tune the detection system so that the false positive rate is acceptable.

Another concern is that many GAs require massive amounts of computation time to create a good set of solutions. This can be reduced by using aggregating functions over Pareto-based approaches for multiobjective problems.

Finally, if all the available statistics are used in classifier generation, then the initial classifiers may describe a very small hypervolume and will not be very good general classifiers. Larger hypervolumes are expected if some genes are not available for evolution and inexact matching. This poses a problem because a difficult decision will have to be made as to which statistics do not deserve to be represented. However, random selection of the statistics to be represented is a feasible approach.

5 Initial Results

This section describes the results that were obtained during initial testing. Testing will continue and will include statistics other than the ones described below as well as other steganography tools. Final results are expected by March 2003.

Self was defined as a collection of 100 clean, 512x512-pixel JPEG images. Wavelet analysis was performed on the self images and 36 separate wavelet coefficient statistics were gathered. The 36 statistics were the mean, variance, skewness, and kurtosis of the coefficients at each subband for all scales. Initial rough analysis to determine which statistics were likely to be the most valuable for detection resulted in the selection of three statistics, the coefficient averages for the three subbands at the first scale, for analysis using the CIS. Different combinations of these three statistics were used by the CIS to produce mature classifiers that detect nonself. These resulting mature classifiers were subsequently deployed against suspect images.

There were four groups of 200 suspect images each. Each group of suspect images was a combination of a common set of 100 clean, 512x512-pixel, JPEG images and varying sets of 100 stego, 512x512-pixel, JPEG images created using Jpeg-Jsteg. The stego images varied in the amount of embedded information to provide a range of detection ease. The embedded information was a random clean image cropped to either 256x256, 128x128, 64x64, or 32x32 pixels.

As expected, the initial detection results varied according to the amount of embedded information and according to the combination of the three wavelet coefficient statistics used. The following table shows average detection rates of stego images with 256x256-pixel, 128x128-pixel, 64x64-pixel, and 32x32-pixel embedded files and average false positive rates:

	Statistics			
	1 and 2	**1 and 3**	**2 and 3**	**1, 2, and 3**
256x256 Detection	88%	84%	85%	87%
128x128 Detection	81%	73%	77%	80%
64x64 Detection	39%	38%	36%	42%
32x32 Detection	24%	28%	21%	28%
False Positives	9%	18%	9%	14%

6 Summary

With this research we will attempt to detect steganography in images that have not been seen before that were created with unknown stego tools. By developing the best available CIS classifiers using statistics from wavelet analysis, we hope to accurately distinguish between clean and stego images. When a CIS with steganography classifiers is used in conjunction with signature-based detectors, the ability to counter the steganography threat will be improved.

References

Anchor, K., Lamont, G., Gunsch, G. (2002). An Evolutionary Programming Approach for Detecting Novel Computer Network Attacks. *Proceedings of 2002 Congress on Evolutionary Computation*, Honolulu, HI, pp. 1618-1623.

Beasley, D., Bull, D., Martin, R. (1993a). An Overview of Genetic Algorithms: Part 1, Fundamentals. *University Computing*, 15 (2):58–69.

Beasley, D., Bull, D., Martin, R. (1993b). An Overview of Genetic Algorithms: Part 2, Research Topics. *University Computing*, 15 (4):170–181.

Buccigrossi, R., Simoncelli, E. (1999). Image Compression via Joint Statistical Characterization in the Wavelet Domain. *IEEE Transactions on Image Processing*, 8 (12):1688–1701.

Coello Coello, C. (1999). A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques, *Knowledge and Information Systems*, 1 (3):269–308.

Farid, H. (2001). Detecting Steganographic Messages in Digital Images. Technical Report TR2001-412, Hanover, NH: Dartmouth College.

Fonesca, C., Fleming, P. (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3 (1):1–16.

Forrest, S., Hofmeyr, S., Somayaji, A. (1996). Computer Immunology. *Communications of the ACM*, 40 (10):88–96.

Fridrich, J., Du, R., Long, M. (2000). Steganalysis of LSB Encoding in Color Images. *Proceedings of the IEEE International Conference on Multimedia and Expo*. 1279–1282. New York: IEEE Press.

Fridrich, J., Goljan, M. (2002). Practical Steganalysis of Digital Images - State of the Art. *Proceedings of SPIE Photonics West 2002: Electronic Imaging, Security and Watermarking of Multimedia Contents IV*, 4675 :1–13.

Harmer, P., Williams, P., Gunsch, G., Lamont, G. (2002). An Artificial Immune System Architecture for Computer Security Applications. *IEEE Transactions on Evolutionary Computation*, 6 (3):252–280.

Hubbard, B. (1996). The World According to Wavelets. Wellesley, MA: A K Peters.

Johnson, N., Duric, Z., Jajodia, S. (2001). Information Hiding: Steganography and Watermarking, Attacks and Countermeasures. Boston: Kluwer Academic Publishers.

Katzenbeisser, S., Petitcolas, F. (2000). Information Hiding Techniques for Steganography and Digital Watermarking. Boston: Artech House.

Luke, S., Spector L. (1997). A Comparison of Crossover and Mutation in Genetic Programming. *Proceedings of the Second Annual Conference on Genetic Programming (GP-97)*: 240–248. Morgan Kaufmann.

Machado, R. (2002). EzStego and Stego. Computer Software. Retrieved February 23, 2002 from http://www.stego.com.

The Math Works. (2001). Wavelet Toolbox User's Guide. Technical Report 2-1. Natick, MA.

Mendenhall, M. (2001). Wavelet-Based Audio Embedding and Audio/Video Compression. MS thesis, AFIT/GE/ENG/01M-18, Graduate School of Engineering and Management, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH.

Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs. Berlin: Springer.

Provos, N. (2002). OutGuess. Computer Software. Retrieved February 23, 2002 from http://www.outguess.org.

Rioul, O., Vetterli, M. (1991). Wavelets and Signal Processing, *IEEE SP Magazine*:14–38.

Upham, D. (2002). Jpeg-Jsteg. Computer Software – Modification of the Independent JPEG Group's JPEG software (release 4) for 2-bit steganography in JFIF output files. Retrieved February 23, 2002 from ftp://ftp.funet.fi/pub/crypt/steganography.

Westfeld, A., Pfitzmann, A. (2000). Attacks on Steganographic Systems - Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools - and Some Lessons Learned. *Lecture Notes in Computer Science*, 1768:61–75.

Williams, P., Anchor, K., Bebo, J., Gunsch, G., Lamont, G. (2001). CDIS: Towards a Computer Immune System for Detecting Network Intrusions. *Lecture Notes in Computer Science*, 2212:117–133.

© 2002 International Journal of Digital Evidence

About the Primary Authors

Captain Jacob Jackson (Jacob.Jackson@afit.edu): Graduated from the University of Florida in 1998 with a B.S. in Electrical Engineering and a B.S. in Computer Engineering and was immediately commissioned as a second lieutenant in the United States Air Force. His first assignment was to Hill Air Force Base, UT where he worked in the Intercontinental Ballistic Missile System Program Office as a Reentry Vehicle Engineer. Following his assignment to Hill, he was selected to attend the Air Force Institute of Technology at Wright-Patterson Air Force Base, OH, where he is currently pursuing his M.S. in Computer Engineering.

Gregg Gunsch, Ph.D. (Lt Col, USAF, ret.) (Gregg.Gunsch@afit.edu): Gregg has a BSEE from the University of North Dakota (1979), a MSEE from the Air Force Institute of Technology (1983) specializing in human and automated information processing, and a Ph.D. in Electrical Engineering from the University of Illinois at Urbana-Champaign (1991) specializing in artificial intelligence and machine learning. He has over fifteen years of experience in developing synergistic computer-human systems through the application of artificial intelligence techniques. He is currently an Assistant Professor of Computer Engineering, bearing the primary responsibility for the information systems security/assurance (information warfare) curriculum at the Air Force Institute of Technology. More at http://en.afit.edu/ggunsch/