2017

The Discrete Spring Transform: An Innovative Steganographic Attack

Aaron T. Sharp
University of Nebraska-Lincoln, atsharp@unomaha.edu

THE DISCRETE SPRING TRANSFORM: AN INNOVATIVE STEGANOGRAPHIC ATTACK

by

Aaron T. Sharp

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Engineering

Under the Supervision of Professor Dongming Peng

Lincoln, Nebraska

October, 2017

THE DISCRETE SPRING TRANSFORM: AN INNOVATIVE STEGANOGRAPHIC ATTACK

Aaron T. Sharp, Ph.D.

University of Nebraska, 2017

Adviser: Dongming Peng

Digital Steganography continues to evolve today, where steganographers are constantly discovering new methodologies to hide information effectively. Despite this, steganographic attacks, which seek to defeat these techniques, have continually lagged behind. The reason for this is simple: it is exceptionally difficult to defeat the unknown. Most attacks require prior knowledge or study of existing techniques in order to defeat them, and are often highly specific to certain cover media. These constraints are impractical and unrealistic to defeat steganography in modern communication networks. It follows, an effective steganographic attack must not require prior knowledge or study of techniques, and must be capable of being implemented against any type of cover media.

Our Discrete Spring Transform (DST) is a highly adaptable steganographic attack that can be applied to any type of cover media. While there are many steganographic attacks that claim to be blind, the DST is one of only a few attacks that does not require training, or prior knowledge of steganographic techniques to defeat them. Furthermore, the DST is one of the only attack frameworks that can be easily tuned and adapted.

In this dissertation, my work on the Discrete Spring Transform will be formally analyzed for its use as an effective steganographic attack. The effectiveness of the attack will be assessed against numerous steganographic algorithms in a variety of

cover media. My research will show that the Discrete Spring Transform is a highly effective attack methodology that can be used to defeat countless steganographic algorithms.

DEDICATION

I would like to thank my advisor Dongming Peng, who has been a phenomenal mentor, and my biggest supporter throughout my graduate career. I would also like to thank my family Tim, Cindy, and Andrew for their continued unconditional support. Thank you.

Table of Contents

List of Figures

Preface

This dissertation contains excerpts from our previous works which appear in the following publications:

1. A. Sharp, Qilin Qi, Yaoqing Yang, Dongming Peng, and H. Sharif. A novel active warden steganographic attack for next-generation steganography. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1138–1143, July 2013

2. A. Sharp, Qilin Qi, Yaoqing Yang, Dongming Peng, and H. Sharif. A video steganography attack using multi-dimensional discrete spring transform. In *Signal and Image Processing Applications (ICSIPA), 2013 IEEE International Conference on*, pages 182–186, Oct 2013

3. Qilin Qi, A. Sharp, Dongming Peng, Yaoqing Yang, and H. Sharif. An active audio steganography attacking method using discrete spring transform. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 3456–3460, Sept 2013

4. A. Sharp, Qilin Qi, Yaoqing Yang, Dongming Peng, and H. Sharif. Frequency domain discrete spring transform: A novel frequency domain steganographic attack. In *Communication Systems, Networks Digital Signal Processing (CSNDSP), 2014 9th International Symposium on*, pages 972–976, July 2014

5. Qilin Qi, A. Sharp, Yaoqing Yang, Dongming Peng, and H. Sharif. Steganography attack based on discrete spring transform and image geometrization. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*, pages 554–558, Aug 2014

6. Aaron Sharp and Dongming Peng. The multi-vector discrete spring transform. *Journal of Information Security and Applications*, 2017. Publication Pending

7. Aaron Sharp and Dongming Peng. An active steganographic attack approach based on perception-preserving discrete spring transform. *Journal of Information Security and Applications*, 2017. Publication Pending

Chapter 1

Introduction

Covert communication is by nature a highly adversarial discipline where one person attempts to communicate securely, and the other attempts to disrupt, prevent, or discover said communication. Digital cryptography and steganography are two such methods for covert communication that have been developed over the last several decades. While the goal of cryptography is to securely protect the delivery of information, steganography's goal is to disguise the existence of information altogether. In this manner, steganography can be an attractive method for covert communication, where unlike cryptography, the entire existence of the communication is concealed. The advantage of steganography over cryptography lies in the fact that cryptographic communication is often very obvious and can be easily prevented or intercepted by a third party, whereas with steganography, the entire existence of information is very difficult to determine [8, 9].

While steganographic techniques have continued to be enhanced in their sophistication and proliferation, steganographic attacks have historically failed to match their pace. In fact, many modern steganographic techniques are engineered to thwart many basic methods of disruption and detection [10]. Therefore, in much the same way that security researchers respond to threats after they are discovered, steganographic attackers must discover techniques to combat them after they are

known. In this regard, steganographic attackers have continually been on the losing side of this battle. Furthermore, in the last several decades, the proliferation of media on the internet has exploded, making analysis, study, and prevention of steganographic communication impractical for a case-by-case basis. In order to truly respond to and thwart steganographic communications, a methodology which is highly adaptable, and capable of blindly attacking steganography is required.

Steganography is considered to be defeated when either the communication is discovered or prevented [8, 9]; in other words, the content of the message does not need to be known to defeat steganography. If follows that the most direct method of attacking steganography is to use an active approach methodology, where an attack attempts to actively disrupt or interrupt communication. Although the vast majority of steganographic attacks rely on passive (steganalysis) methods, which analyze a media to assess the likelihood it contains steganograhic data, these techniques are impractical for defeating steganography. The reason is that passive detection always requires some training or analysis of existing steganographic methods to be effective. Even passive techniques which claim to be blind require unrealistic training or machine learning processes [11–14]. In contrast, active attack methodologies can be implemented against any type of cover media or steganographic algorithm. Why then have active approaches been overshadowed by passive techniques? The reason is that active approaches are considered destructive, unpredictable, and difficult to tune or adapt. For example, while Stirmark [15, 16] (a widely used active steganographic framework) is widely used as an active attack framework (typically for testing robustness of steganographic techniques), virtually no researchers have given it serious consideration as a steganographic attack for the aforementioned reasons. It follows that an active

attack methodology that is non-destructive, predictable, and effective is required to realistically defeat steganography.

The Discrete Spring Transform (DST) that we have developed is an active, highly adaptable, non-destructive steganographic attack. Unlike other active attack methodologies, the DST has been engineered to attack any number of steganographic algorithms in virtually any type of digital media [1–7]. The basis for the DST lies in exploiting a fundamental constraint of all steganographic algorithms, which is, numeric stability of a digital media is required for successful steganographic communication. In other words, the numeric values of a digital media are required to remain somewhat constant in order for a media to be successfully used for steganographic communication. While this initially seems like a reasonable constraint, given that changes in a media's numeric values seem likely to distort or alter the quality of the media, countless research has been produced which indicates that this is not true [17]. By exploiting this weakness, we have developed an attack that is efficient, effective, and adaptable to effectively defeat steganography.

In this Dissertation, my work on the Discrete Spring Transform (DST) will be formally described, modeled, and shown to be an effective steganographic attack. A methodology for tuning and adapting the DST will be formally described and applied to defeat numerous types of steganographic techniques in a variety of cover media. The results of my research will show that the DST is a next-generation, highly adaptable steganographic attack, capable of defeating even the most advanced steganographic schemes in highly distributed environments.

Chapter 2

Motivation

Covert communications have been in use for hundreds of years and continue to evolve today. Some of the most prolific moments in history have involved uncovering or intercepting secret communications. Julius Caesar was thought to have used ciphers to communicate with his legions in ancient times [18]. The German Army engineered the Enigma cipher machine as a highly robust way for the third Reich to communicate, and the cracking of the Enigma in World War II was arguably a major turning point for the allies [19]. The invention of public key cryptosystems transformed network security and ushered in a new era of secure communications [20]. Countless other equally profound moments have involved the use of covert and secure communication systems. Despite the wide variety in these scenarios and the sophistication of the techniques involved, the one constant is the adversarial nature of secure communication. This classic dilemma is illustrated nicely by the prisoner's problem. In the prisoner's problem, two prisoners are attempting to communicate securely by passing messages to each other through a warden [21]. The communication between the prisoners is considered secure if the true content of the message cannot be discovered by the warden [21]. Furthermore, if the prisoners want to disguise the existence of the message altogether, then the communication is only considered secure if the

warden cannot determine with certainty that the message contains any covert communication [21]. It follows that all secure communications have at least three parties involved: The sender, the recipient, and the attacker.

While the sender and recipient have many methods of communicating covertly, what happens if the physical delivery of the information is compromised? Any intelligent attacker would quickly be able to realize certain communication streams are using certain blatant covert communication methods and disrupt or otherwise prevent the successful delivery of this information. How then can two individuals communicate without having their communication channel interrupted? One solution to this problem is to disguise the entire existence of covert communication altogether. This practice is referred to as steganography, and involves transporting information in a manner that is seemingly innocuous [8,9]. Digital steganography typically involves encoding information within a digital communication medium with a large data capacity, such as a digital audio, image, or video source, or any other large benign file [8,9]. Unlike cryptography, where an attacker can be reasonably certain that a certain communication stream contains covert data, with steganography, the distinction is almost impossible. An attacker cannot simply disrupt or prevent all content from being transported, as the vast majority of data is benign. In this regard, steganography is an attractive method to distribute covert data on communication networks. As a result, preventing secret communication when steganography is involved is a much more difficult problem to address if one simply wishes to end the covert communication.

Over the years, the proliferation of personal computers have made covert communication through digital cryptography and steganography very simple for virtually anyone to implement and use. Anyone with a computer and access to cryptographic or steganographic software can potentially become a sender

in the prisoner's problem using any number of freely available software suites. Furthermore, the widespread availability of communication and media outlets on the internet has also made it simple to proliferate covert communications to virtually any place at any time. While nearly all secure communications have an innocent purpose, there are nefarious individuals that will seek to use covert communication for their own ends. Over the years there have been relatively few concrete discoveries of steganography being used in the wild, but those that have been found have had disturbing implications. In [22], a terrorist cell was found using steganography to encode the plans of an upcoming attack within a video. Another similar example revealed that intelligence agents had been using steganographic software to encode information [23]. A more benign but equally important example found that a software company had been secretly encoding screenshots generated by their software [24]. The real threat of steganography in modern communication networks is that the content is often untrusted and unregulated, allowing anyone to encode and hide malicious information anonymously and alongside the rest of the benign information. For this reason, the warden, or attacker serves an important purpose in the ecosystem for secure and covert communication networks.

While interrupting communication via cryptography is a simple manner (the attacker simply disrupts the communication channel), defeating steganography is a much more difficult and profound problem for attackers to address. Over the years, countless techniques have been established which can uncover numerous types of stego-data and algorithms in a variety of media [8, 9], however, these attacks suffer from a fundamental issue: they require knowledge of the algorithm they intend to defeat. In other words, attacks against covert communication methodologies have involved a discovery phase, where a new methodology is discovered for

secure communication, followed by an attack phase, where individuals attempt to find methods of disrupting or defeating this newly found communication system. While this is typical of most security fields, it is simply impractical to fully address the issue at hand. Any clever steganographer could monitor current attack schemes, and modify their techniques accordingly. Furthermore, this modification is often trivial for a steganographer to make. In fact, making slight changes to certain mechanics of an encoding algorithm can bypass certain detection schemes entirely. In order to truly disrupt steganography, an attacking method that is blind and does not require study of steganographic schemes is required. In this manner, prevention can be implemented against any digital media that is considered suspect. While many have attempted to develop blind attack methodologies [11–14], nearly all existing approaches require a training phase, which again, requires knowledge of the steganographic attacks they intend to defeat.

It follows that attackers have yet to discover a truly blind methodology for attacking steganography. In this regard, those wishing to use covert communication for nefarious purposes need only monitor the current steganographic attack methodologies and modify their algorithms accordingly. Clearly, this cat and mouse game is a losing battle for attackers, as discovery is often the most difficult component of developing an attack. To truly address the threat of covert communication using steganography, an attack methodology that is highly adaptive, blind, and efficient is required. This is the motivation behind our Discrete Spring Transform, which is an attack that seeks to be truly blind, highly adaptable, and efficient in attacking modern digital steganography.

Chapter 3

Background

There has been extensive research in both steganographic algorithms and corresponding attacks over the last several decades. I will attempt to briefly review modern steganographic algorithms and attacks. The intent of this review is not to provide a comprehensive list of existing techniques and attacks, but rather to highlight the various types of methods that exist.

An important preface for this section is in relation to the definition of steganography versus watermarking. It is commonly accepted by researchers that watermarking always constitutes a positive or non-nefarious goal whereas steganography is not always benign [8, 9, 25, 26]. This distinction has led to branches in research where attackers generally ignore watermarking in lieu of steganography. Fundamentally however, both watermarking and steganography hide information within cover media, regardless of the intention of the encoder. For this reason we treat watermaking and steganographic techniques identically since they both fundamentally accomplish the same end-goal. In the context of steganographic attacks, it is important to recognize that the intention of the steganographer is unknown. The positive or nefarious intentions of a steganographer cannot be understood by an attacker, and assessing this is difficult and outside the scope of this research.

3.1 Steganographic Techniques

The body of research that has been conducted in steganographic techniques is overwhelming (see [8, 9]), especially when compared with the existing research of steganographic attacks. Despite the fact that steganography is inherently an adversarial game, steganographers have an exceptionally simpler task than attackers, if for no other reason that steganographers have a larger body of research at their disposal.

Providing a comprehensive analysis of existing steganographic techniques is not feasible for this dissertation, however, I will attempt to highlight important techniques and categories of methods.

3.1.1 First Generation Steganography - Least Significant Bit

Least significant bit steganography involves modifying the raw bit representation of a media to encode information. This type of steganography is applicable for any type of media that has a digital representation and is often the simplest type of steganography to implement. Methods that encode information in audio, images, and video are prevalent [8, 9, 27–33] and despite the fact that such methods are often simple to uncover or destroy, researchers continue to find more sophisticated methods of using LSB techniques.

These types of techniques were groundbreaking at the time of their discovery but have since been considered antiquated due to the fact they often leave very telling signs of manipulation for attackers to discover. Despite this fact, LSB steganography continues to be researched to this day and new techniques are constantly emerging which offer various tradeoffs for capacity, robustness, and ease of implementation.

3.1.2 Second Generation Techniques - Transform Domains

Almost immediately after LSB steganography had begun to emerge, steganographers began encoding information in alternative representations of cover media. It was discovered that encoding information within alternative transform domains, such as the frequency domain, can produce robust, high capacity techniques that are more difficult to uncover by attackers. Such techniques are more difficult to detect because the information within the media is spread more evenly and can observe cover media statistics more easily than LSB techniques.

Some of the most prolific steganographic techniques utilize transform domains, including F5, Outguess, and JSteg [34, 35]. In fact, one of the most cited and most prolific methods of encoding information within images involves using Spread-Spectrum techniques to encode a spreading-sequence in the DCT coefficients of an image [26]. Despite the fact these methods ushered in a new generation of steganographic techniques, they often suffer from the same problems that LSB methods do since they can be easily discovered if they fail to observe known statistical metrics of the cover media. In fact, one can abstract most LSB techniques to an alternative domain quite easily. The major contribution of these methods is the realization that a cover media can contain information in an alternative domain. It follows that for a given transform domain N, one can simply encode information in the $(N+1)^{th}$ domain as long as they observe the statistical properties of that domain.

3.1.3 Advanced Techniques - Robustness Against Attack

Modern steganographers have begun to realize that an algorithm is ineffective if it is not robust against attacks and have started implementing techniques which

are immune to active attack methods. The majority of these techniques attempt to embed information within target components which are thought to be critically important to the perception of the media. In this manner, information is embedded within components of a media that must remain unharmed to be properly perceived, thus in theory keeping stego-data safe from destruction.

The majority of such techniques have been directed at audio and image-derived steganography. In terms of image steganography, Rotation Scaling and Translation (RST) resistant techniques are the most prolific types of techniques. These methods are capable of resisting distortion introduced by basic spatial operations [10, 36–41]. These techniques are some of the first within image-steganography that have directly addressed the issues presented when an active attacker is attempting to thwart a steganographic scheme.

Likewise within audio-steganogrpahy several approaches have been made to deter the effect of Time-Scale Modification (TSM) on embedding techniques [42–45]. Such techniques are capable of resisting distortion that might be introduced through alterations to an audio sequence's time-scale. Again, these techniques operate by encoding information in a way that it is always encoded within critically important sections of the audio-sequence. The concept is that altering the time-scale where the embedding takes place should significantly degrade the audio quality.

Often these attack-resisting techniques are concerned more strictly with maintaining integrity than covering their existence, however, it is not difficult to imagine a scenario where a steganographer combines algorithms which are robust against attacks with algorithms that avoid detection in order to form a method that is both highly robust and transparent. For this reason, we believe the next generation of steganographic techniques will fall into this category of implementation.

3.2 Passive Steganographic Attacks

Despite the fact that digital steganography has been heavily researched for the last several decades, steganographic attacks have historically lagged behind techniques. The reason for this is simple, it is difficult and sometimes impossible to anticipate or counterattack the unknown. As a result, the vast majority of attacks fall into the passive attack model, where a media is scanned or otherwise checked for the existence of stegnographic data.

In general, passive steganographic attacks are referred to as steganalysis which is the study of a cover media to determine if it contains any suspect or hidden information [8]. The goal of steganalysis is not to discover the actual hidden information within a cover media, but rather to determine the existence of the stego-data, since steganography is considered defeated if the existence of the information is known.

3.2.1 First Generation Steganalysis - Statistical Modeling

The first generation of steganalysis attacks is based on the concept of determining a set of statistics or known base metrics for certain types of cover media and comparing suspect media against these statistics. The attacks described in [46–51] are typical examples of how known statistics of a cover media can be used to uncover the existence of hidden stego-data. The concept behind these attacks is always that a steganographic technique will alter the normal statistics of a media in a telling way. These attacks are called first generation attacks because they are highly specialized to certain types of cover media and corresponding attacks. Despite the fact these techniques can be quite successful, they are unrealistic in practice. As a steganographer becomes aware of existing attacks, they can

effectively thwart the attack by adhering to the expected statistics or metrics of the attack. Likewise, a steganographer can simply shift their algorithm to a different domain of the cover media, for example encoding information within the frequency domain, and defeat most attacks that look for statistics within the spatial domain.

3.2.2 Advanced Steganalysis - Machine Learning

Despite the fact that it is exceptionally difficult to detect algorithms that have not been studied, several steganalysis techniques have attempted to resolve this shortcoming using various methods. Techniques that are based on Support Vector Machines (SVM) have been employed which attempt to classify stego-data using machine learning methods [11–14]. In this manner, an SVM-based system may be trained to recognize media that are likely to contain steganographic data. In theory this overcomes the limitations of passive steganography since a machine can be trained to recognize stego-media. Other researchers have proposed alterations or enhancements to the learning method with varying degrees of success [52, 53] but at the core of each technique the concept of machine-learning is employed.

Once again, these methods suffer from the possibility that a steganographer will simply design their algorithm to avoid detection by these specific attacks. Furthermore, providing a sufficiently large set of media to train the machine-learning algorithm typically requires knowledge of existing steganographic techniques, which defeats the entire purpose of a blind general purpose attack.

3.3 Active Steganographic Attacks

As previously stated, the majority of steganographic attacks are passive in nature. Despite this, numerous approaches have been discovered to attack steganogra-

phy in an active way. Most approaches that utilize this methodology introduce distortions to cover media in a certain manner. The reason these approaches are effective is that a cover media's numeric values are often not strictly important in the perception of the media [54–57]. In other words, a cover media's numeric values can be changed or distorted to a certain degree without affecting the quality of the media too severely. However, a cover media's numeric representation is extremely important when encoding stego-data and even slight distortions can render a steganographic scheme ineffective [58–61].

Despite the fact that the active attack model was postulated several decades ago [21, 62] there are surprisingly few implementations that have been discovered. Active attacks are in fact so rare that many researchers simply model attacks as random noise, where several researchers have discussed the effect of combating steganography using noise [63]. Likewise, various other methods have been proposed which seek to eliminate steganography using distortion or spatial transforms [15, 16], but these techniques are often limited to specific cover media and the effectiveness of the attacks is not well understood.

Other researchers have proposed specific implementations of active attacks that are targeted at specific types of cover media that have been shown to be effective at removing steganographic data [64–66]. These approaches are certainly in the right direction for an active steganographic framework but still lack the generality and adaptability that is required of a modern active attack.

Lastly, one unique approach that has been proposed is attacking steganography at the network layer to combat the covert channel on the internet [67]. This approach is novel in that the authors proposed an attack that was not strictly targeted at the application layer. However, the attack is still rather primitive and is not intelligently targeted at cover media but at internet traffic as a whole.

3.4 Steganographic Attack Frameworks

A steganographic framework is a collection of tools or attacks that can be used to discover or remove steganography within a cover media. Most frameworks that exist today are simple collections of existing attacks and leave the decision of how to attack or disrupt a media to the end-user of the framework.

3.4.1 Stegdetect

Stegdetect is a component of the Outguess framework [68, 69] that is a suite of steganographic tools that can be used to discover the existence of JPEG steganography within cover media. Stegdetect essentially is an aggregate tool suite that attempts to uncover cover media that have been encoded using JSteg, JPHide, or Outguess techniques for encoding information [68]. The tool itself is passive in nature and simply aggregates existing steganalysis methods. This framework is unsuitable for the needs of a modern steganographic framework since it exclusively utilizes passive techniques to detect cover media within JPEG images.

3.4.2 Reference Framework

The framework proposed in [70] suggests a method for discovering steganography within images using image references. The concept behind the algorithm is assembling a collection of non-encoded images and using the reference colors within said images to compare against suspected cover media. The methodology is unrealistic for a modern steganographic framework since it requires assembling a large library of reference images and likewise uses passive steganalysis techniques.

3.4.3 Stirmark

Stirmark has arguably been the most prolific steganographic framework to emerge from steganographic research in the last two decades [15, 16]. Stirmark contains a collection of active attacks that introduce distortions to cover media in various ways. These distortions exploit the fact that steganographic schemes require a certain stability in the numeric values of a cover media. Despite the fact that Stirmark has been strongly accepted within the research community (at the time of this writing over 75 publications within IEEE Xplore have cited Stirmark since 2005), it still lacks many components of a modern attacking framework, including the ability to adapt to various types of cover media (Stirmark is predominately concerned with attacking image and audio-derived media), and the ability to be used on a massive scale (Stirmark requires a lot of manual intervention and decision making for the end-user to effectively use it).

Stirmark has certainly paved the way for modern steganographic attack frameworks but there are significant improvements that must be made before it can be used to combat steganography on the internet and many of the attacks within the framework are based on intuition rather than concrete results.

3.5 Summary

As evident, steganography has been heavily researched both in terms of techniques that encode and hide information and techniques that attack or attempt to remove hidden information. Despite the fact that steganographic techniques continue to increase in their sophistication, attacks continue to lag behind. Several groups of researchers continue to make efforts to remedy this shortcoming within the field, however, current techniques are still limited for several reasons.

Despite the overwhelming body of research into steganalysis, such techniques are still flawed in their ability to quickly adapt to new steganographic schemes, and even techniques which claim to implement blind attacks require a learning process. Several researchers have begun to understand the importance of active attacks and frameworks, however, these efforts are few when compared to passive methodologies.

Chapter 4

An Effective Steganographic Attack

Steganography, in its most simplistic definition, is a method with which to disguise the existence of information. Like cryptography, it is often used to send information covertly or in a manner that information is not easily intercepted by an attacker. It is therefore quite reasonable to describe cryptography and steganography using traditional communication network paradigms. In this manner, digital steganography resembles a communication network, where the stego data is the signal and the digital media is the channel. Although an attacker has any number of methods at their disposal to defeat a communication network, most network attacks are based on the concept of jamming the channel by introducing noise or distortion. Despite this realization, most steganographic attackers rely on passive approaches for discovering the existence of steganography. The reason being that one wishes to avoid introducing unnecessary distortions or negative impacts to the attacked media. However, discovery is insufficient to prevent the actual communication from occurring, and even once the communication is identified, the optimal response mechanism for dealing with the communication is unclear. As a result, an effective steganographic attack must take a more active approach to defeating steganography in order to directly prevent steganographic communication while minimizing unnecessary disruptions.

4.1 Steganography Numeric Stability

Consider a steganographic function $S(X, D) = Y$ that accepts a cover media X and stego-data D and produces an encoded stego-media Y which contains D. Likewise, consider its corresponding inverse function $S^{-1}(Y) = D$ which accepts an encoded stego-media Y and produces the encoded stego-data D. For S to maintain proper communication during transmission, the Bit Error Rate (BER) of the scheme must remain above a certain threshold β (β may vary depending on the scheme in place and other error-prevention mechanisms such as Forward-Error Correction).

Let $\hat{Y} = Y + \epsilon = S(X, D) + \epsilon$ be a transmitted cover media, where ϵ is an error or noise signal. For Y to be properly received the relationship:

$$\frac{\sum S^{-1}(\hat{Y}) - S^{-1}(Y)}{N} < \beta \tag{4.1}$$

must be preserved, where N is the size of D.

4.2 Performance

When attempting to blindly defeat steganography, an attacker has no definitive knowledge of the steganographic algorithm in use nor the information that is being encoded. Knowledge of this information would mean that the steganographic algorithm has already been defeated. As such, in a typical scenario of blind steganographic attacks, the steganographic algorithm S and the data D are considered completely unknown to an attacker. We therefore must define a metric of describing how effective a steganographic attack is against a steganographic algorithm without having any knowledge of S and D.

4.2.1 Steganographic Embedding Set

Conceptually, all steganographic algorithms embed information by altering values in a digital media. For a given media X and data set D, S must always produce a media Y using a known and consistent method. That is, for distinct S, X, and D, Y must also be distinct to be properly received. Therefore, there is always a set of digital values in Y that are used to carry steganographic data. By introducing errors into these information carriers in Y, one may defeat a steganographic algorithm in the same manner a communication network may fail under a poor Signal-to-Noise Ratio.

For a given steganographic algorithm S, let $E(S, X, D)$ be the *Embedding Set* of digital values for S, X, and D. We define the *Embedding Set* as the set of values in X that carry steganographic information for a given S, X, and D. Although E may vary significantly for different S, X, and D we know that the following properties must hold true for all E:

- $\forall e \in E(S, X, D),\ e \in X$
- $\forall s \in S,\ x \in X,\ d \in D\ \exists\ E(x, s, d)\ s.t.\ |E(s, x, d)| \geq 1$
- $d_1, d_2 \in D,\ d_1 \neq d_2,\ E(S, X, d) \neq E(S, X, d)$

This derivation assumes that for a fixed S and D, the size of E remains fixed as well, and henceforth we denote the size of the embedding set E as ρ.

4.2.2 Quantization-based Embedding

Along with the selection of the embedding set, a steganographic algorithm S must also alter the values within the embedding set in a distinct manner. My research has focused this derivation on steganographic algorithms which utilize

quantization embedding. A quantization embedding method is any manner of altering a value within the embedding set E such that the value assumes one of several fixed quantization levels. Given S, X, $d \in D$, α a quantization strength, and A quantization levels, the embedding process for $E(X, S, d)$ may be described as follows:

$$\forall e \in E(X, S, d), \hat{e} = e + \frac{n\alpha}{2A} \tag{4.2}$$

where $n = -A, -A+1, \ldots, A-1, A$. n is chosen for each $d \in D$ and $e \in E$ and may be selected randomly depending on the steganographic scheme in place. For instance, if one considers a method which uses a spreading-sequence or random code to embed information within each embedding set, the selection of n is essentially randomized.

Using an embedding scheme E and a quantization-embedding method, an encoded stego-media Y may be found as follows:

$$Y = \begin{cases} x & x \notin E \\ \hat{x} & x \in E \end{cases} \tag{4.3}$$

Using this definition of Y, we can approximate the BER of S for Y and $\hat{Y}$ (where $\hat{Y}$ is the attacked version of Y) as:

$$BER(Y, \hat{Y}) \approx \sum_{y \in Y} \frac{(\hat{y} - y)e(y)}{\alpha N} \tag{4.4}$$

where $N = |D|$ and

$$e(x) = \begin{cases} 0, x \notin E \\ 1, x \in E \end{cases} \tag{4.5}$$

4.2.3 Performance Metric

Once again, certain components of equation 4.4 are unknown to the attacker such as the embedding set E, the quantization strength and size α and A, and the message length N. However, unlike prior definitions of the performance of a steganographic scheme S, equation 4.4 is written entirely in terms of Y, which is critical as the basis of a performance metric.

As a result, we introduce a metric which we call the steganographic performance factor P that allows an attacker to approximate the BER of S. The performance factor $P(Y, \hat{Y})$ can be found as:

$$P(Y, \hat{Y}) = \sum_{y \in Y} \frac{(y - \hat{y})p(y)}{\alpha N} \tag{4.6}$$

where $p(y)$ is a probability density function approximating the location of embedding values in X, such that:

$$p(y = 1) = \frac{\rho}{|X|} \int_0^1 f(y) d_x \tag{4.7}$$

and $|X|$ is the size of the cover media and ρ the size of the embedding set. $f(x)$, should be a probability density function thought to best approximate the location of embedding values in X, most likely this will be a uniform distribution. The probability distribution is scaled by $\frac{\rho}{|X|}$ to account for the possibility that a single value in X may hold steganographic data for multiple embedding sets in E.

Despite the fact that it is impossible for an attacker to know α, N, and $p(y)$,

these values can be approximated or assumed as a worst-case scenario. For instance, if the attacker assumed that each value in X was within an embedding set in E, and that α and N (the embedding strength and message length) were both very large, the attacker would attempt to heavily distort each value in X with severity. This is, of course, a poor assumption, since a steganographer would attempt to make the existence of stego-data as transparent as possible, but the point remains that these constants may be tuned by the attacker depending on how suspect the cover media appears to be.

Using approximations of α, N, and $p(y)$, an attacker must observe the following inequality to defeat a steganographic scheme:

$$\sum_{y \in Y} \frac{(y - \hat{y})p(y)}{\alpha N} \geq \beta \tag{4.8}$$

where β is the minimum performance score thought to defeat a steganographic algorithm. Typically, even small β values are sufficient to defeat a steganographic algorithm, but the worst case scenario would attempt to reach a factor of 0.5, which is somewhat analogous to a BER of 0.5, indicating the message is completely unrecoverable.

4.3 Quality

A steganographic attack must also preserve the perceptual characteristics of a media to be successful. That is, an attacked digital media must remain perceptually identical to another but is not required to retain numeric characteristics. Perceptual identity is difficult to assess and is often described on a per-media basis as will be elaborated in the following sections.

4.3.1 Perceptual Identity

A digital media's perception is not strictly tied to its numeric representation. This is an obvious observation if one simply considers the various ways human beings may perceive objects. Subtle changes in light, saturation, color, etc, all often go unnoticed by human observers, yet said changes can have a drastic impact on the raw quantitative measure of a media, that is, its numeric representation.

We define a function S as the perceptual similarity of two media X and Y as follows:

$$S(X, Y) = \tau \tag{4.9}$$

where τ is a numeric value between 0 and 1, where 1 indicates the media are completely perceptually identical, and 0 that they are not perceptually identical. A value between 0 and 1 simply means that the media has lost some of its perceptual quality. This loss in perceptual quality typically signifies distortion that impacts the perception of the media, or various other operations that may also negatively impact its quality. Historically, S has been measured in terms of the raw numeric discrepancies between two media via Mean Squared Error.

4.3.1.1 Peak Signal to Noise Ratio

Peak Signal to Noise Ratio (PSNR) measures the raw numeric differences between two discrete signals and is the most well-established method for assessing the quality of discrete signals. The PSNR [71] for two discrete signals X and Y is defined using Mean Squared Error (MSE) as follows:

$$\frac{1}{N} \sum_{i=0}^{N-1} [X(i) - Y(i)]^2 \tag{4.10}$$

PSNR is then given as:

$$PSNR(X,Y) = 20log_{10}(MAX_X) - 10log_{10}(MSE) \tag{4.11}$$

where MAX_X is the maximum possible value for X and the units of the PSNR are in Decibels. Typically, a PSNR of over 30 dB indicates that the signal has maintained an acceptable quality, though this is highly specific to different types of media and acceptance levels for quality.

4.3.1.2 Structural Similarity Index

Current research into media quality analysis has yielded various methods of measuring a media's quality that are agnostic to the numeric discrepancies of the media. Despite the wide set of algorithms that exist, the most commonly used and widely accepted methodology is the Structural Similarity Index [17]. Although this approach is specific to 2-dimensional signals (for example, images), the approach may be extrapolated to other dimensions.

We define the Structural SIMilarity Index (SSIM) for two images X and Y as follows [17]:

$$SSIM(X,Y) = [l(X,Y)^{\alpha} * c(X,Y)^{\beta} * s(X,Y)^{\gamma}] \tag{4.12}$$

where $l(x,y)$, $c(x,y)$, and $s(x,y)$ are comparison operations, such that l compares the luminance, c the contrast, and s the structure of the two images. These functions are defined defined as follows:

$$l(x,y) = \frac{2\mu_x * \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_2} \tag{4.13}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{4.14}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{4.15}$$

where μ_x, σ_x, and σ_{xy} are specified in terms of a media x of size N as follows (note that $x(i)$ is the media's intensity at the i^{th} position):

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x(i) \tag{4.16}$$

$$\sigma_x = (\frac{1}{N-1} * \sum_{i=1}^{N}(x(i) - \mu_x)^2)^{\frac{1}{2}} \tag{4.17}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x(i) - \mu_x)(y(i) - \mu_y) \tag{4.18}$$

The constants, C_1, C_2, C_3, α, β, and γ are used to fine-tune the SSIM and are typically defined as $C_1 << 1$, $C_2 = (K_1L)^2$, $C_3 = \frac{C_2}{2}$, $\alpha = \beta = \gamma = 1$, where $K_1 << 1$ and L is the dynamic range of the pixels (0 - 255).

Thus for two 2D media X and Y to be perceptually identical, they must have a SSIM greater than a threshold τ where the relationship $SSIM(X,Y) \geq \tau$ must be preserved. Replacing constants, C_1, C_2, C_3, α, β, and γ with the accepted constants yields the following equation for SSIM:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.19}$$

4.3.2 Perceptually Identical Media

Using S as a basis, we can formally describe the restrictions for a steganographic attack which will produce two perceptually identical media. In this manner, the attack will alter a cover media's numeric representation while still maintaining an acceptable media quality that is perceptually identical to the original cover media. We can therefore state that for a steganographic attack $A(X) = \hat{X}$ to maintain perceptual identity of a media X, the following inequality must be observed:

$$S(A(X), X) \geq \tau \tag{4.20}$$

where $A(X) = \hat{X}$ is the attacked version of X.

4.3.2.1 Mean Squared Error Perceptually Identical Media

Recall that the PSNR measures the quality between two media X and Y. For two media X and Y to be perceptually identical, they must have a $PSNR$ greater than a certain threshold, denoted here as τ. Therefore, the following relationships must be preserved:

$$20log_{10}(MAX_X) - 10log_{10}(\frac{1}{N}\sum_{i=0}^{N-1}[X(i) - Y(i)]^2) \geq \tau \tag{4.21}$$

Substituting Y for $A(X)$, we find that the following inequality must be preserved:

$$20log_{10}(MAX_X) - 10log_{10}(\frac{1}{N}\sum_{i=0}^{N-1}[X(i) - A(X(i))]^2) \geq \tau \tag{4.22}$$

4.3.2.2 SSIM Perceptually Identical Media

Recall that the SSIM index measures the structural similarity between two media X and Y. Thus for two media X and Y to be perceptually identical they must have an $SSIM$ index greater than a certain threshold, denoted here as τ. Thus the following relationship must be preserved:

$$\frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \geq \tau \tag{4.23}$$

In general, we can assume that a successful attack will retain the global characteristics of a media. Thus to simplify this derivation we make the assumption that $\mu_x \approx \mu_y$ and $\sigma_x \approx \sigma_y$. It follows that the inequality in equation 4.23 can be rewritten as:

$$\sigma_{xy} \geq \tau\sigma_x^2 \tag{4.24}$$

Substituting the equations for σ_{xy} and σ_x defined in equations 4.18 and 4.17 respectively, we find the following inequality must be preserved:

$$\sum_{i=1}^{N} y(i)(x(i) - \mu_x) \;\geq\; \tau[(N-1)\sum_{i=1}^{N}(x(i) - \mu_x)^2]^{\frac{1}{2}} + \sum_{i=1}^{N}\mu_x(x(i) - \mu_x) \tag{4.25}$$

Thus an attacked image $y(i) = A(x(i))$ must maintain the following inequality to produce a perceptually identical image:

$$\sum_{i=1}^{N} A(x(i))(x(i) - \mu_x) \geq \tau[(N-1)\sum_{i=1}^{N}(x(i) - \mu_x)^2]^{\frac{1}{2}} + \sum_{i=1}^{N}\mu_x(x(i) - \mu_x) \tag{4.26}$$

The importance of Equation 4.26 is that perceptual identity of a given media can be directly assessed via easy to compute properties of the image and tuned via τ.

Chapter 5

Fundamental DST Attack

Our first implementation of the Discrete Spring Transform was against image derived media using algorithms which stretch and compresses portions of an image or video file non-linearly. The concept was derived using existing image transformation techniques which can quickly and efficiently resize images and videos according to various parameters. The choice of image and video-derived media was due to the prevalence and proliferation of image-derived steganography as well as the wide array of tools that can manipulate image and video-based media.

5.1 DST for Image-Derived Media

We will now rigorously define the DST for image-derived media. In order to realize the DST, the digital image is first interpolated into a continuous 2-D image, which can be expressed as:

$$\hat{A}(x,y) = A(x,y) * W_L(x,y) \tag{5.1}$$

where $A(x,y)$ is the $M \times N$ original image, and W_L is the interpolation window kernel. In this paper, the 3rd-order Lanczos window kernel of 1-D form can be

expressed as:

$$w(x) = 3 * \frac{sin(\pi x) sin(\frac{\pi x}{3})}{\pi^2 x^2}, |x| = 0, 0 < |x| < 3, 3 \leq |x| \tag{5.2}$$

and the 2-D window kernel is given as:

$$W_L(x, y) = w(x) \cdot w(y) \tag{5.3}$$

Next, $\hat{A}(x, y)$ is re-sampled using variable sampling rates which can be expressed as:

$$A'(x, y) = \hat{A}(S(x), Q(y)) \tag{5.4}$$

where $S(x)$ and $Q(y)$ are random curves representing the variable sampling rates. For example, as shown in Figure 5.1, $S(x)$ maps $x_i \rightarrow x'_i$, which makes the locations of the re-sampling points from $\hat{A}(x, y)$ irregular. It can be shown that if $S(x) = x$ and $Q(y) = y$ then the re-sampled image A' will be identical to A. Thus, in order to make the re-sampled points disordered while keeping A' the same size as A, $S(x)$ and $Q(y)$ should be monotonically increasing and the relationship $S(M-1) \leq M-1,\ Q(N-1) \leq N-1$ must be observed.

It follows that this definition of DST can be applied to a variety of domains and media, not exclusively to image-derived media. In this aspect, the cover media previously defined as A will take the form of another type of media or steganographic domain. The definition of DST still holds but is applied in a different manner to the cover media.

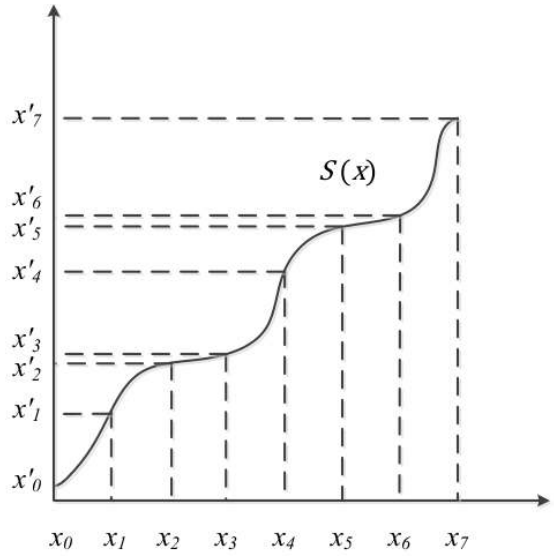


Figure 5.1: Spring Variable Sampling Rate Curve

5.2 DST Sample Attacks

In order to better illustrate applications of the DST we will describe some concrete instances of attacks that are simple to conceptualize. The following attacks are not derived from any existing steganographic attacks but are simply derived DST attacks that we have conceived to best represent typical DST applications; these attacks are original techniques specific to DST and as such we have coined several terms to describe them (pinch, spatial warp, and dimensional). For these examples, we will focus on image-derived cover media, but as previously stated the cover media can be diverse.

5.2.1 Pinch Attack

A pinch attack is the simplest example of a DST attack, where the term pinch derives from the concept of compressing a portion of a cover media, that is, pinching it. In a pinch attack, a given cover media is transposed into its 2-D

representation (or possibly as a sequence of 2-D representations) and a given section of the image is compressed or reduced in size, while the remaining section of the media is expanded to fill the reduced space. While this attack is extremely simple it can often prove effective at defeating a variety of steganographic schemes as the statistics of the image are distorted in a manner that makes preservation of stego media difficult. The attack directly distorts the image reducing the quality but depending on the parameters of the pinch these effects can be negligible.

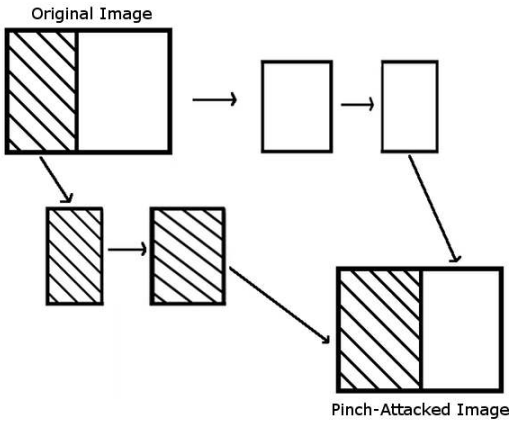


Figure 5.2: Pinch Attack

5.2.2 Spatial Warp Attack

A warp attack is a super-set of the pinch attack and describes any spatial operation that may be applied to a 2-D representation of cover media, that is, a specific type of spatial warp or distortion is applied to a cover media. Such attacks can use any variety of spatial transforms to attack the stego media without severely degrading the cover-media's quality.

5.2.3 Dimensional Attack

A dimensional attack describes skewing or altering a cover media in a given dimension, hence the name. For example, Spatial Warp attacks and its child attacks (pinch attack) are considered attacks in 2-D space, where the media is altered within 2-dimensional space. Depending on how a cover media is defined it may be susceptible to attacks in multiple dimensions. For example, audio may often be described using multiple channels, where each channel may be considered a possible dimension for attack. Similarly, video may be described as a sequence of 2-dimensional frames, where time may be considered a third dimension for attack. Attacking a cover media in unconventional domains (such as channels for audio, or time for video) may produce some excellent active warden attacks, in that they can be very successful at destroying the stego media while preserving the cover media's quality. In the future we hope to further explore unique dimensional attacks and provide some concrete examples of possible DST for these domains.

5.3 Steganographic Attacks

To demonstrate the effectiveness of the DST attack we have chosen to attack two different next-generation steganographic algorithms: Motion Vector Steganography and RST-Resilient Steganography. We have chosen to attack these algorithms as they both utilize techniques which are considered cutting-edge and robust against traditional steganographic attacks.

5.3.1 Motion Vector Steganography

The Motion Vector Steganography works by modifying the motion vectors of a video stream to hide data, where many techniques have been proposed which

embed information in this domain [32, 33, 72, 73]. The algorithm is effective since slight alterations to motion vectors are virtually undetectable through traditional image-based steganographic attacks. The algorithm is robust against compression or other problems which typically obscure or distort the hidden steganographic data, making it a prime target for an active warden attack. In fact, currently the only proposed attack that has been observed in literature is a passive warden attack which is highly specific to motion vector steganography [74].

5.3.2 RST-Resilient Steganography

As previously described, many types of RST-resilient algorithms exist, however, we have chosen to focus on an algorithm which encodes data in a normalization domain, specifically the algorithm described in [75]. This type of RST algorithm is the most typical example of how RST can be implemented and has been proven robust against common signal processing attacks and geometric distortions. The strength of RST algorithms is that they are capable of resisting active steganographic attacks as opposed to most algorithms which merely attempt to protect against passive techniques to disrupt data. Thus RST algorithms are prime targets for active-warden attacks.

5.3.3 Discrete Spring Transform Attack

For Motion Vector and RST resilient algorithms the DST attack is very similar, the primary difference being that the attack must be applied frame-by-frame to a video sequence in the case of Motion Vector steganography. In the case of Motion Vector steganography, the attack is implemented by first encoding a video stream with the Motion Vector steganographic algorithm described in [32]. The DST is then applied to each frame of the video. For this specific attack, we arbitrarily

chose to implement a pinch transform of each frame, where a certain section of the frame is squeezed, and the remaining section of the frame is stretched. The size and compression ratio of the pinch attack is swept against various values, where the size of the pinch selection and compression ratio dictate how much of frame is pinched, and how much this collection is compressed respectively.

The resulting frame is slightly distorted but retains the properties of the original frame, such as the size. A 'pinch' is used simply because it is easy to implement and apply to individual frames of a video, however, any number of DST algorithms could be applied. This 'pinch' will clearly distort the frame, and in fact is the reason that the hidden message will be destroyed. Despite this, the transform is essentially invisible to the naked eye and does not significantly distort the video, which will be verified by comparing the PSNR before and after the transform. After the DST is applied the resultant video is decoded and the BER is determined for the extracted message. For the RST-Resilient attack an image was first encoded using the algorithm described in [75]. The image was then attacked using a pinch transform, which was identical to the pinch used to attack each frame of the Motion Vector algorithm. The image was then decoded and the BER and PSNR were determined in the same manner as the Motion Vector attack.

Chapter 6

Multi-Dimensional DST Attack

The second iteration of the DST attack exploited the fact that multiple dimensions of a media can be attacked simultaneously. For example, certain steganographic algorithms may encode information within a spatial domain, whereas other media may encode information within the time domain. Attacking multiple dimensions simultaneously makes the DST more powerful since it can likewise attack different types of steganographic algorithms simultaneously.

6.1 Video Steganography

While video steganography is a relatively new steganographic medium, there have been some interesting schemes proposed which encode information in multiple domains of video sequences. Most of these techniques fall into one of three categories: 2-dimensional encoding, 3-dimensional encoding, and multi-dimensional encoding.

6.1.1 2-Dimensional Video Steganography

2-Dimensional video steganography refers to any techniques which may be used to encode information within individual frames of a video sequence using image-

based steganography where example algorithms may be found in [76–78]. Since these techniques only operate 2-dimensionally within individual frames of the video sequence, the term 2-dimensional steganography is appropriate. There is nothing gained over normal image-based steganography using these techniques as the strength of the algorithms are not enhanced when applied to video.

6.1.2 3-Dimensional Video Steganography

3-dimensional video steganography refers to techniques which attempt to encode information using a third dimension of the video sequence, such as time or motion vectors.

With time-based steganography, information may be spread in time by altering only certain frames, or sections of frames within a video sequence using image-based steganography. The advantage to this approach is that only a fraction of the possible frames and data are encoded, making steganographic attacks difficult since most of the video sequence will not contain any steganographic data. As a result, many steganographic attacks that take advantage of predefined statistics within image or video sequences would likely fail since the encoded video largely retains the same metrics as the original sequence.

Motion vector steganography encodes information within the motion vectors of a video sequence typically by intercepting the motion estimation block (as found in popular video compression algorithms) and altering motion vectors in a certain way [32, 33, 72, 73]. This technique utilizes motion between frames which is also considered a 3-dimensional medium for encoding. This attack is unique in that it takes advantage of a video-specific medium to encode information, meaning that image-based steganographic attacks are inadequate to defeat this type of steganography. Currently, the only observed attacks in literature are passive

warden attacks that are specific to motion vector steganography [79,80].

6.1.3 Multi-Dimensional Video Steganography

Multi-dimensional video steganography refers to a combination of 2-Dimensional and 3-Dimensional video steganography. Multi-dimensional steganography can simultaneously encode information in both the 3D and 2D sections of video, resulting in an extremely large capacity for steganographic data. In fact, often both techniques can be encoded independently of each other, meaning it is possible to encode two different sequences of information in two different domains of the video simultaneously. Figure 6.1 shows a block diagram of how 2D and 3D video steganography can both be applied to a video sequence. In this sample scheme, each frame of the video is encoded using standard image-based steganography (this frame is called the IFrame). Next, the next frame in the sequence (called the PFrame) is used to perform motion estimation from the IFrame. The PFrame is altered using motion-vector steganography to encode information. The cycle is then repeated by advancing the sequence using the PFrame as the new IFrame. The result of this type of encoding is that there is no current steganographic attack that can simultaneously address the 2D and 3D encoding in the video sequence. For this reason, we have chosen to attack multi-dimensional video steganography using the multi-dimensional DST to show how this attack can simultaneously defeat two different types of steganography schemes.

6.2 System Architecture and Methodology

We will now formally describe the Discrete Spring Transform for video steganography and some sample applications for specific types of cover media. The definition

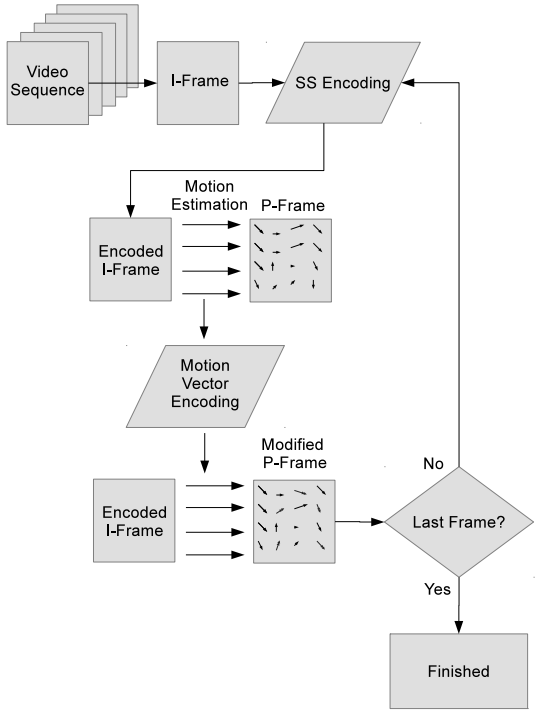


Figure 6.1: Video Steganography Encoding

of the Discrete Spring Transform is independent of any specific steganographic algorithm and can be applied to any type of cover media in n-dimensional space.

6.2.1 Discrete Spring Transform

Let C be an n-dimensional cover media defined as:

$$C = F(x, y, z, \ldots) \tag{6.1}$$

where

$$x, y, z, \ldots \in Z \tag{6.2}$$

and the number of parameters in F(x, y, z, ...) is n.

The Discrete Spring Transform for a cover media C and attacked cover media $\hat{C}$ may be described as follows:

$$C = F(x, y, z, \ldots) \rightarrow AF(\lfloor ax \rfloor, \lfloor by \rfloor, \lfloor cz \rfloor, \ldots) = \hat{C} \tag{6.3}$$

and A, a, b, c, $\ldots \approx 1$ and are defined as:

$$\begin{aligned} A &= f_1(x, y, z, \ldots) \\ a &= f_2(x, y, z, \ldots) \\ b &= f_3(x, y, z, \ldots) \\ c &= f_4(x, y, z, \ldots) \\ \ldots &= f_n(x, y, z, \ldots) \end{aligned} \tag{6.4}$$

The strength of the Discrete Spring Transform lies in the definition of $f_n(x, y, z, \ldots)$, which we define as any non-linear and time-variant function. Unlike simple RST transforms, the non-linearity of the DST is applied to each dimension of the image.

6.2.2 DST for Image Media

Define an M x N pixel gray-scale image I as a cover media $I = F(x, y)$, where the number of pixels in x is M, and the number of pixels in y is N.

The DST is then realized as:

$$I = F(x, y) \rightarrow AF(\lfloor ax \rfloor, \lfloor by \rfloor) = \hat{I} \tag{6.5}$$

where A, a, b are defined as:

$$\begin{aligned} A &= f_1(x,y) \\ a &= f_2(x,y) \\ b &= f_3(x,y) \end{aligned} \tag{6.6}$$

and $f_n(x,y)$ is any non-linear time-variant function.

6.2.3 DST for Video Media

Define an M x N x F video (consisting of a sequence of F M x N gray-scale images) as a cover media $V = F(x,y,z)$, where the number of pixels in x is M, the number of pixels in y is N, and the number of frames is F.

The DST is then realized as:

$$V = F(x,y,z) \rightarrow AF(\lfloor ax \rfloor, \lfloor by \rfloor, \lfloor cz \rfloor) = \hat{V} \tag{6.7}$$

where A, a, b, c are defined as:

$$\begin{aligned} A &= f_1(x,y,z) \\ a &= f_2(x,y,z) \\ b &= f_3(x,y,z) \\ c &= f_4(x,y,z) \end{aligned} \tag{6.8}$$

and $f_n(x,y,z)$ is any non-linear time-variant function.

6.3 Video Steganography Attack

As other steganographers have observed, video steganography is fast becoming an interesting new steganographic medium which has enormous capacity compared with traditional steganographic cover mediums [32, 33, 72, 73, 76]. For this reason,

we have chosen to apply the multi-dimensional DST attack to video steganography. We have chosen to attack a scheme which encodes information in multiple steganographic domains of the video sequence, using image-based steganography and motion-vector steganography. Figure 6.1 describes the process of encoding information in the video sequence where information is encoded 2-dimensionally within individual frames of the video, as well as 3-dimensionally within the motion vectors of the video. We believe this scheme represents a robust system that would be exceptionally difficult to combat using existing steganographic attacks.

The attack will utilize 2D and Time (3D) DST attacks to combat the multi-dimensional video steganography scheme. Figure 6.2 describes the process of attacking the video sequence as follows: First, the video sequence is decomposed into a train of 2D images or frames. Next each frame of the sequence is attacked using the 2D DST transform. Lastly, this resultant sequence is attacked using the Time (3D) DST attack. The semantics of the 2D and Time (3D) DST attacks are described in the following sections.

6.3.1 2D DST Attack

The 2-dimensional DST attack has been previously described in [1], where the attack was applied to individual frames of a video sequence. The 2D DST attack can more generally be defined as an operation which will spatially distort media that can be expressed 2-dimensionally using a nonlinear spatial transform. Various algorithms may be applied which fit the criteria of a 2D DST attack, however, for simplicity we will focus on attacking the media using a 'pinch' attack, where individual sections of two-dimensional media are stretched and other sections are compressed. The net effect of this nonlinear spatial attack is that the media retains some slight distortion but the attack is effective in destroying most hidden stegano-

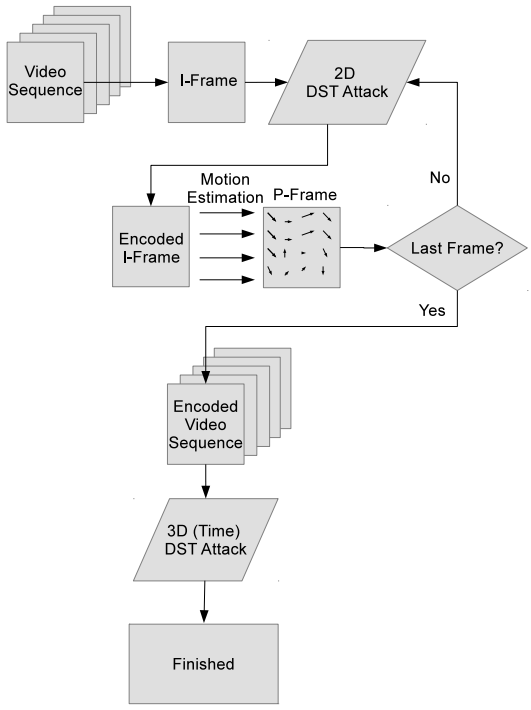


Figure 6.2: DST Video Steganography Attack

graphic data while maintaining an acceptable PSNR. This attack has been proven to be effective at combating complicated cover media such as video sequences, and will be part of the multi-dimensional Spring attack.

6.3.2 DST Time Attack

The DST Time attack is in principle identical to the 2D DST attack but is implemented in the third dimension of the steganographic media rather than the second dimension. It is understood that this attack can only be applied to those types of cover-media which exhibit at least three dimensions, such as video sequences. For a video sequence, this attack can be thought of as affecting the time or framerate, hence the title DST Time attack. Figure 6.3 describes the process of a simple DST Time attack, where a video sequence is first arbitrarily split into two

video sequences. Next, each of these sequences is stretched or compressed via 3-dimensional interpolation in the time dimension. The result is that the number of frames in one sequence is decreased while the number of frames in the other sequence is increased. The resulting sequences are then combined to form a video sequence that has the same number of frames as the original sequence. This attack will be applied as part of the multi-dimensional Spring attack.

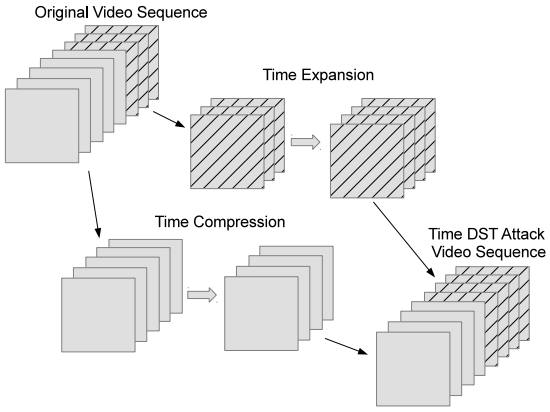


Figure 6.3: DST Time Attack

Chapter 7

Domain-based DST Attack

In the same way that steganographers realized there are advantages to encoding information within alternative domain representations of a media, the next evolution of the DST attack was that the attack could be applied to an alternative domain as well. Attacking alternative domains of a media, such as the frequency domain, distributes the attack more evenly, which improves efficiency and quality by distributing the distortions across the media instead of localizing them to certain spatial regions.

7.1 System Architecture and Methodology

We now formally describe the Frequency DST (FDST) attack for image-derived cover media, using the Fourier Transform as the reference frequency domain. The FDST can be applied to other types of cover media and frequency domains as well, however, we restrict the definition to images using the Fourier transform for simplicity.

7.1.1 Frequency-based DST for Image-derived media

Let $C = c(x, y)$ be an M x N pixel gray scale image, where the number of pixels in x is M and the number of pixels in y is N.

We define the 2D Fourier transform of C, $\hat{C}$ as:

$$\hat{C} = F(C) \rightarrow F(w_1, w_2) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} c(i, j) e^{-i2\pi(\frac{w_1 i}{M} + \frac{w_2 j}{N})} \tag{7.1}$$

We next select the mid-range frequency components of $\hat{C}$, $M_{\hat{C}}$ using parameters $\gamma_1, \gamma_2, \delta_1, \delta_2$ as follows:

$$M_{\hat{C}} = \{F(w1, w2) \mid \gamma_1 < w_1 < \gamma_2,\ \delta_1 < w_2 < \delta_2\} \tag{7.2}$$

We select the mid-range frequency components as most steganographic schemes encode information here to avoid distorting the cover media, and likewise we also wish to avoid distorting the cover media too severely. Note however that the choice of γ and δ is left to the attacker and may be chosen however is most appropriate.

$M_{\hat{C}}$ must next be partitioned into a set of blocks $B(w_1, w_2)$ with a randomly selected block size. The selection of these blocks is randomized to attempt to attack the encoded information with as much irregularity as possible. In other words, most steganographic schemes employ some method of error correction, which assumes that errors are applied with some uniformity. The randomized selection of these blocks attempts to defeat such correction techniques by introducing as much non-linearity as possible.

Define $P_{M_{\hat{C}}}$ as the set of all blocks B in $M_{\hat{C}}$ as follows:

$$P_{M_{\hat{C}}} = \{B(w_1, w_2) \mid B \in M_{\hat{C}}\} \tag{7.3}$$

The partitioning of $\hat{C}$ is used to account for possible irregularities in the selection of M_C. For each block $B \in P_{M_{\hat{C}}}$ we perform the 2D DST transform [1] to find the DST attacked block $\bar{B}$ as:

$$\bar{B} = DST_{2D}(B) = A * B(\lfloor aw_1 \rfloor, \lfloor bw_2 \rfloor) \tag{7.4}$$

where A, a, b are randomized non-linear time-variant functions.

Once the 2D DST attacked blocks are found the image is reconstructed using these attacked blocks to obtain the FDST attacked image, inverting steps (such as the Fourier transform) where necessary.

7.1.2 Frequency-based DST Algorithm

The FDST is easily described algorithmically, where figure 7.1 demonstrates the algorithm in pseudo code (note that FFT and $IFFT$ refer to Fast Fourier Transform and Inverse Fast Fourier Transform respectively).

1: **procedure** FREQUENCY_DST(C, γ, δ)
2: $\hat{C} \leftarrow FFT(C)$
3: $M_{\hat{C}} \leftarrow mid(\hat{C}, \gamma, \delta)$
4: $P_{M_{\hat{C}}} \leftarrow \{B \mid B \in rand\_partition(M_{\hat{C}})\}$
5: **for all** $B \in P_{M_{\hat{C}}}$ **do**
6: $\hat{B} \leftarrow DST(B)$
7: **end for**
8: **return** $IFFT(\hat{C})$
9: **end procedure**

Figure 7.1: Frequency DST Algorithm

Where figure 7.2 shows how the frequency domain of a cover media is masked to find the mid-range frequency components and figure 7.3 shows how the mid-range frequency band is partitioned into sub-blocks for the attack.

```
1: procedure MID(Ĉ, γ, δ)
2:     m ← {}
3:     for all c(w1, w2) ∈ Ĉ do
4:         if (γ1 < w1 < γ2) & (δ1 < w2 < δ2) then
5:             m ← m ∪ c(w1, w2)
6:         end if
7:     end for
8:     return m
9: end procedure
```

Figure 7.2: Mid-Range Frequency Component Selection

```
1: procedure RAND_PARTITION(M_Ĉ)
2:     B ← {}
3:     x ← 0
4:     y ← 0
5:     while x < w1 do
6:         while y < w2 do
7:             x ← x + rand()
8:             y ← y + rand()
9:             b ← M_Ĉ(x, y)
10:            if b ∈ M_Ĉ then
11:                B ← B ∪ b
12:            end if
13:        end while
14:    end while
15:    return B
16: end procedure
```

Figure 7.3: Random Partitioning Algorithm

7.2 Frequency Domain Discrete Spring Transform Attack

For the Frequency DST attack (FDST) we concentrated our efforts on the Fourier transform of the cover media, however, most frequency domain transforms could be interchanged (for example Discrete Cosine Transform) since they are so similar. The main revisions of the FDST from a traditional DST attack involve determining where the attack is more effectively concentrated in the frequency domain. As previously stated, the mid-range components of the FFT are typically least affected

by distortion, in fact, this is where most steganographic schemes embed information. With this premise we choose to attack the mid-range frequency components of the FFT cover media. As the DST is more easily implemented by attacking square or rectangular regions (since it typically requires interpolation), it is simpler to partition the mid-range frequency components into randomized rectangular sub-sections. This is done by masking off the mid-range components of the cover media and partitioning them into arbitrary sized rectangular regions. After this is accomplished, the DST is applied normally to each rectangular region using a 'pinch' transform as described in [1]. The pinch parameters for each DST are randomized to provide maximum disruption of cover media and apply a more uniform distortion to the DST. The FFT cover media is then reassembled using the attacked regions and transformed back to the spatial domain.

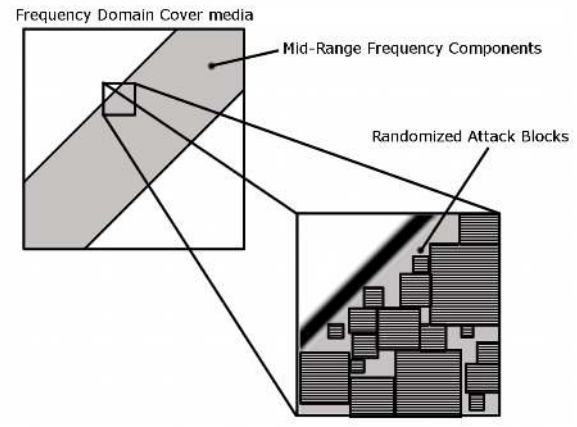


Figure 7.4: FDST Attack Diagram

As evident, there are many portions of the algorithm used within this attack where the parameters can be tuned for either strength or quality of the cover media. The most obvious choices are the size and position of the mid-range frequency components and the size of the rectangular partitions.

Chapter 8

Multi-Vector DST Attack

A significant improvement to other DST implementations is the development of a generalized DST framework to attack a media using multiple simultaneous attack vectors while maintaining the media's perceptual identity. Attacking a media with multiple simultaneous vectors drastically improves the performance of an attack since the attack can be targeted against a variety of steganographic algorithms that may encode data in different vectors of a media.

8.1 Perceptually Faithful Only DST

The basis of the Multi-Vector DST (MV-DST) attack is the realization that two images may maintain perceptual identity without maintaining numerical identity. As previously described, steganography can be considered a form of covert communication where the stego-media is a carrier or channel for hidden information. In order to maintain communication using steganography, the channel or stego-media must maintain a certain Signal-to-Noise Ratio (SNR) to be properly received.

Utilizing our definitions of performance and perceptual identity from 4.2.3 and 4.3.1 respectively, an algorithm which maintains perceptual identity while

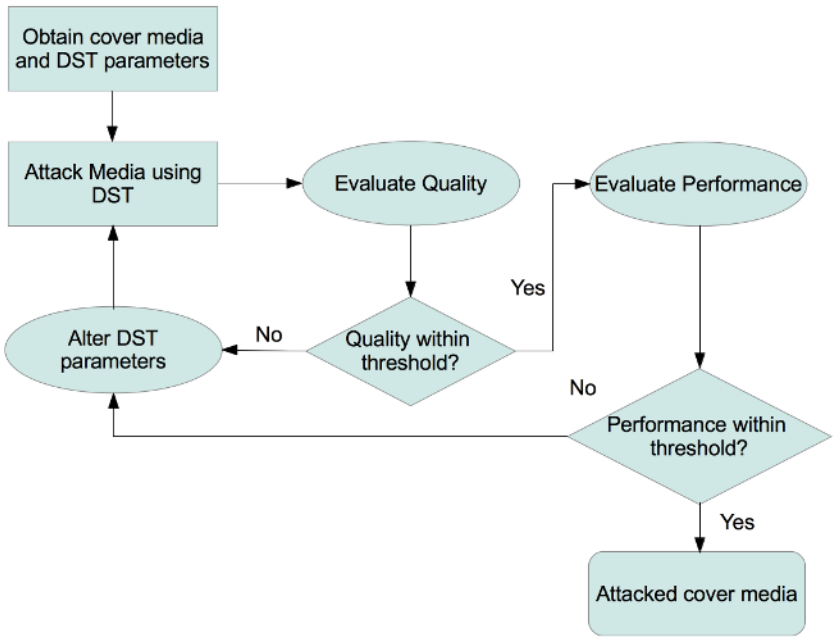


Figure 8.1: PFO DST Attack Diagram

maximizing attack performance is defined in figure 8.1.

Direct implementation of this algorithm is impractical as it requires being able to compute the performance of the attack, which will not be known to an attacker. However, this algorithm is very useful when combined with estimations of performance in terms of other known DST properties. This relationship will be elaborated in the formal MV-DST methodology.

8.2 MV-DST Framework

Previous implementations of the Discrete Spring Transform were implemented as a singular attack vector [1–5], where a specific domain of a cover media is disrupted in a specific manner. For instance, several DST implementations displace vectors using interpolation-based techniques within spatial or frequency domains

[1, 2, 4]. These DST implementations in essence implement the attack in a very specific, singular vector, meaning the directionality of the approach is fixed. A DST implementation which can be applied in multiple simultaneous vectors of any domain would be highly advantageous for an attacker to achieve maximal adaptability and flexibility in how the disruption is applied. Furthermore, a formally-defined DST would allow an attacker to more succinctly describe and tune the characteristics of the disruption.

8.2.1 Multi-Vector Directional Discrete Spring Transform Attack

Let C be a digital cover media with N dimensions and discrete intensity levels ranging from 0 to α where each value in C takes the form of $c(x_1, x_2, \ldots, x_n)$.

To perform the Multi-Vector Discrete Spring Transform attack we first define a Spring Mesh Φ as follows:

$$\Phi(X) = \Phi(x_1, x_2, ..., x_n) = (x_1 + \phi_1, x_2 + \phi_2, ..., x_n + \phi_n) \tag{8.1}$$

where ϕ_x is a random value such that $-\frac{1}{2} < \phi_x < \frac{1}{2}$ and the size of Φ is $2N$.

The details of selecting an appropriate Φ are up to the attacker and constraints and considerations for selection of Φ are further discussed in 8.2.3.

Now, Φ is used to determine the continuous Spring Mesh mapping G of the media C as follows:

$$G = \begin{cases} g(0) = c(0) \\ g(\sigma_{x_1}(x_1, x_2, \ldots, x_n), \sigma_{x_2}(x_1, x_2, \ldots, x_n), \ldots, \sigma_{x_n}(x_1, x_2, \ldots, x_n)) = \\ c(x_1, x_2, \ldots, x_n)(1 + 2\Phi_{n+1}(x_1, x_2, \ldots, x_n)) \end{cases} \tag{8.2}$$

where

$$\sigma_\gamma(x_1, x_2, \ldots, x_n) = \gamma + \int_{[0,\gamma]} \Phi(x_1, x_2, \ldots, x_n) d\mu_(\gamma) \tag{8.3}$$

In this manner, the Spring Mesh mapping G contains values of C that have been displaced and scaled from their original position and intensity.

Since G is continuous, the position of values within G do not necessarily coincide with the original discrete positions in C. In order to translate G back to the original discrete domain of the media C, an inverse function G^{-1} (referred to as the Spring normalization function) is required which is defined as follows:

$$\begin{aligned} G^{-1} = g^{-1}(x_1, x_2, \ldots, x_n) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \\ \sum_{X_1=x_1-\beta}^{x_1+\beta} \sum_{X_2=x_2-\beta}^{x_2+\beta} \cdots \sum_{X_n=x_n-\beta}^{x_n+\beta} \frac{g(X1, X_2, \ldots, X_n)}{\| < X_1, X_2, \ldots, X_n > - < x_1, x_1, \ldots, x_n > \|} \end{aligned} \tag{8.4}$$

where $x_1, x_2, \ldots, x_n \in C$. G^{-1} is essentially just the weighted average of points within a block of β in the Spring Mesh mapping G. The points are weighted using the squared Euclidean distance between the target point and points found within G. The choice of β is again up to the attacker and a discussion of impacts for the choice of β is further discussed in 8.2.3.

The result of $G^{-1}(G)$ is the MV-DST attacked media.

8.2.2 Attack Properties and Characteristics

The Discrete Spring Transform is comprised of several important properties: continuity, elasticity, and reactivity. These properties directly impact the quality of the cover media and the performance of any steganographic carriers within the media.

8.2.2.1 Continuity

The continuity of the MV-DST refers to how smooth, or continuous, changes in the resultant DST-encoded media are. Consider that a media which exhibits sharp, rigid, or discontinuous areas would likely result in a media with low perceptual quality.

The continuity of the Γ^{th} dimension of DST-encoded media Δ_Γ is defined in terms of the Spring Mesh Φ as follows:

$$\Delta_\Gamma = \sum_{\gamma=1}^{\|\Gamma\|} \frac{|\Phi(\gamma) - \Phi(\gamma - 1)|}{2 \parallel \Gamma \parallel} \tag{8.5}$$

The smoother Φ is, the greater Δ is and the more continuous the DST-encoded media will be. When considering that media components are directly displaced by Φ, the smoother Φ is the less rigid or discontinuous the DST-encoded media will be.

8.2.2.2 Elasticity

The elasticity of the MV-DST is a measure of how alterations to a particular section of a media impact other neighboring regions of the media. The basis of the DST is that of altering a media in a manner that introduces highly-localized distortions that impact neighboring regions proportionately. Consider that if a particular section of a media is enlarged or stretched, neighboring regions are shrunk or scaled to maintain the size and average characteristics of that section. This produces an encoded media that is not simply an affine or scaling operation, but one that is non-linear and maintains global characteristics of a media. In fact, elasticity is one of the most important characteristics of the DST.

The elasticity of the DST is directly impacted by the choice of β when perform-

ing the inverse Spring Mesh mapping. As β approaches the size of the cover media, each point within the Spring Mesh mapped media G impacts each resultant point in the MV-DST-encoded image. Conversely, a small β will produce a media where alterations to particular regions may have little to no effect on neighboring regions. As a result, the elasticity ϵ is proportional to β and is defined as:

$$\epsilon = \frac{\beta}{\parallel \Phi \parallel} \tag{8.6}$$

where $\beta \geq 1$.

8.2.2.3 Reactivity

Reactivity refers to how strongly components of the DST-encoded media are displaced or scaled. A reactivity score of 0 would indicate that the DST-encoded media is identical to the original, or in other words, was not attacked. The reactivity of the γ^{th} dimension of ϕ, ρ_γ, is defined as follows:

$$\rho_\Gamma = \sum_{\gamma=0}^{\|\Gamma\|} \frac{|\Phi(\gamma)|}{2 \parallel \Gamma \parallel} \tag{8.7}$$

We define the reactivity in terms of each dimension of the Spring Mesh since it is not obvious how to compare the reactivity for intensity versus position values. It is also likely that the attacker may choose to have a different reactivity score for positional versus intensity values, and the computation of ρ_γ allows individual tuning of media reactivity.

8.2.3 Attack Considerations

An attacker whom carefully considers the effect a given Spring Mesh Φ will have on continuity, elasticity, and reactivity will produce a more optimal attack than

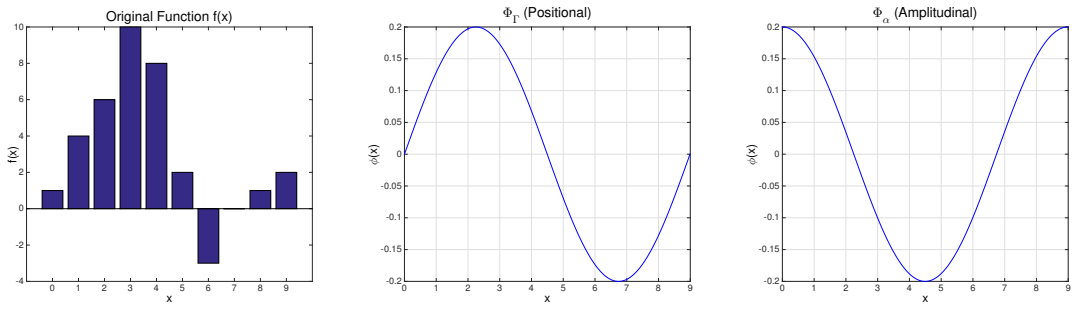
a naive attacker. Recall from [1] that the goal of the DST attack is to introduce highly localized, non-linear variations to the media. An attacker that maximizes continuity, elasticity, and reactivity while observing an appropriate perceptual quality will maintain these important DST characteristics without disturbing the quality of the media.

Since Φ is simple to construct and the continuity, elasticity, and reactivity are simple to observe and compute, tuning Φ is a relatively straightforward process. In general, there are no strict limitations placed on the selection of Φ, but the following relationships are suggested from observed results:

- $\sum \phi \approx 0$
- $2 \leq \beta \leq \frac{\Gamma}{4}$

The summation of ϕ (which is distinct from the reactivity), should be approximately equal to 0 in order to preserve, as closely as possible, the original scale of the media. Likewise, the selection of β has been shown to be most effective when it is not too small or large, where observed values of 2 and $\frac{\Gamma}{4}$ make effective lower and upper limits. Attackers may choose to tune or alter ϕ as appropriate for the attack at hand. Likewise, ϕ may also vary significantly if an alternate attack domain, such as the frequency domain, is used as the target domain.

Most steps of the MV-DST are computationally simple to perform, however, the normalization phase can be very computationally intensive, since it must search the Spring-Mesh g for indices that are at a distance of β. The computational complexity of the normalization process can be substantially improved if the indices in g are quantized or rounded to fixed positions. This quantization process essentially eliminates the search that needs to be performed since the indices can be arranged after they are quantized. Although the quantization process reduces the accuracy



(a) Original Function (b) Phi Gamma (c) Phi Alpha

Figure 8.2: Original Function and Φ

of the normalization process, the increased computational performance is often worth this decrease in accuracy. Likewise, the attacker can choose between the complexity and accuracy of the normalization process to suit their needs.

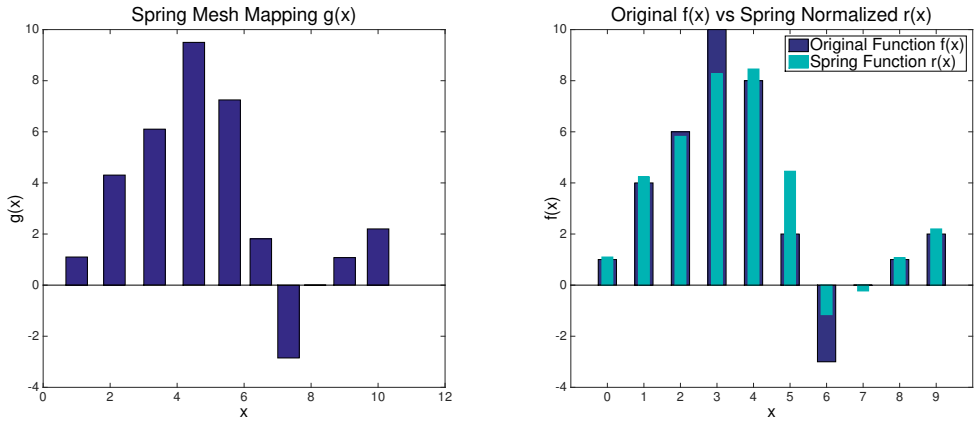
8.2.4 A Concrete Example

8.2.4.1 1-Dimensional Example

Consider a 1-dimensional function $f(x)$ as depicted in figure 8.2a and a corresponding Spring-Mesh Φ as depicted in figures 8.2b and 8.2c.

Φ displaces positional indices and scales amplitudinal values as described in 8.2. Since $f(x)$ is two-dimensional, the x indices of the function $f(x)$ are scaled by Φ_Γ. Consider the third position of $f(x)$, where $x = 2$ and $z = 0$. The Spring-Mesh mapped index for $g(x)$ is equal to $\sigma_x(x) = x + \sum_0^2 \Phi(x, 1) = 2 + 0.3255 = 2.3255$ and the amplitudinal value is $g(2.3255) = f(2) * (1 + \Phi(2, 2)) = 6 * (1 + 0.0174) = 6.1044$. The operation is repeated for each value in $f(x)$ forming the Spring-Mesh mapping $g(x)$ as depicted in 8.3a.

After the Spring Mesh mapping is found, it needs to be normalized to adhere to the dimensions and discrete points of the original function $f(x)$. As described in 8.4, each discrete point in $f(x)$ is iterated and the normalized value is calculated by



(a) Spring Mesh Mapping (b) Original vs Spring Comparison

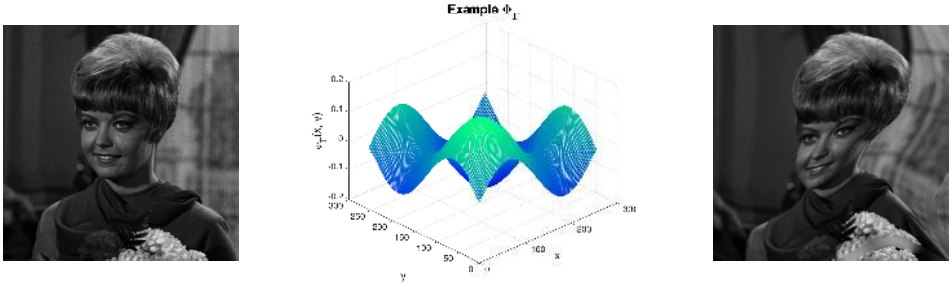
Figure 8.3: Spring Mesh and Normalization Comparison

computing the weighted average of points within β. The metric used for weighting the points found within β is the squared Euclidean distance. For example, consider the second discrete point in $f(x)$, where $x = 1$. If we let $\beta = 2$, then we find 3 points in $g(x)$ where $x = 1, 2.1286, 3.3255$ and $g(x) = 1.1, 4.3064, 6.1042$ respectively. We next compute the weighted average of these points using the squared Euclidean distance as the metric as follows:

$$f(1) = \frac{\frac{g(1.1)}{<1.1,1>} + \frac{g(2.1286)}{<2.1286,1>} + \frac{g(3.3255)}{<3.3255,1>}}{< 1.1, 1 > + < 2.1286, 1 > + < 3.3255, 1 >} = \frac{\frac{1.1}{0.01} + \frac{4.306}{1.274} + \frac{6.104}{5.41}}{0.01 + 1.274 + 5.41}$$

8.2.4.2 Image Example

To further demonstrate how the MV-DST may be applied to a more complex function, a gray-scale image (which is essentially a 2-dimensional discrete function), is attacked using the MV-DST. Figure 8.4 demonstrates the results of performing the MV-DST against a gray-scale image, where figure 8.4a shows the original



(a) Original Image (b) Φ Γ (c) MV-DST Image

Figure 8.4: MV-DST Image Example

image, figure 8.4b the Spring Mesh Φ, and figure 8.4c the result of the MV-DST. In this scenario, the reactivity ρ of the Spring Mesh is exceptionally high for demonstration purposes. For the purpose of defeating steganography within a cover media, the reactivity would likely be much lower to maintain appropriate perceptual quality.

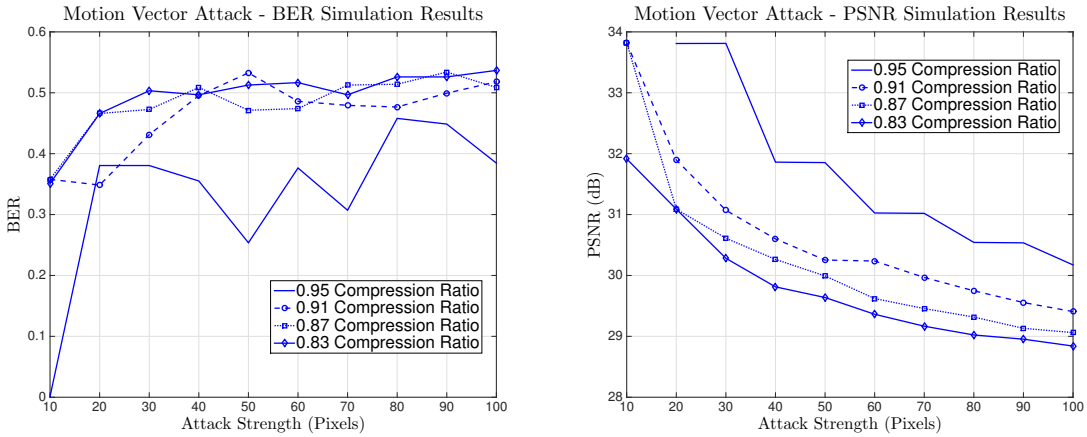
Chapter 9

Results

As evident, the Discrete Spring Transform has been used in numerous applications to defeat a wide array of steganographic algorithms in a wide array of cover media. I will now describe simulation results we've obtained when applying the DST to various types of steganographic algorithms.

9.1 Fundamental DST Attack

To demonstrate the Fundamental DST attack, a video steam consisting of 100 frames was encoded with a random bit string using the Motion vector steganographic algorithm in [32], where the algorithm alters motion vectors within the video stream to hide the message. Likewise an image was encoded in a similar manner using the RST-Resilient steganographic algorithm in [81]. A pinch transform was then applied to each frame of the video (or image, in the case of the RST algorithm), where the size of the pinch was swept from from 0 to 100 pixels. We have dubbed the size of the pinch the width, which is the size of the image (measured in pixels) that will be attacked. Next, several different compression ratios were chosen for the attack, ranging from 0.65-0.95, where the compression ratio is the percentage of the pinch that was compressed, that is, the compression



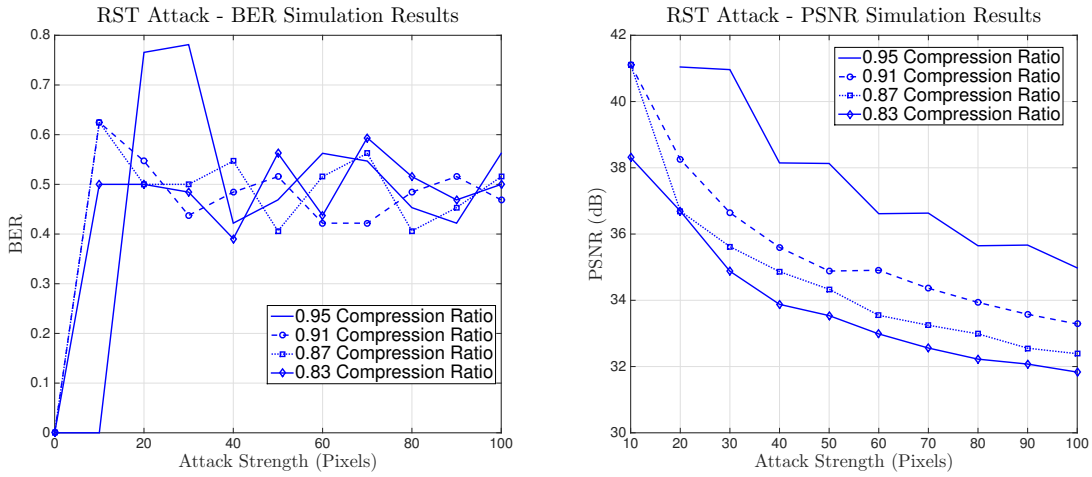
(a) BER vs Pinch Width

(b) PSNR vs Pinch Width

Figure 9.1: Motion Vector Attack

ratio is defined as the size of the resultant pinch versus the original pinch. Thus a compression ratio of 0.95 indicates that the original pinch is compressed to 0.95 of its original size. It follows that the larger the pinch, and the lower the compression ratio the greater the strength of the attack. Since the attack relies only on some basic 2-dimensional (or possibly 3-dimensional in the case of RGB image) spatial transforms via interpolation of matrices we observe that the attack has the potential to be implemented efficiently on a variety of platforms. At a pinch width of 0 pixels the DST encoded frame was identical to the original frame or image, meaning the PSNR was infinite and the BER was identical to that of the original encoded image. Figures 9.1 and 9.2 demonstrate how the BER and PSNR of the encoded messages changes with the pinch width.

It is easily observable through figures 9.1a and 9.2a that the BER of the encoded message increases dramatically to values between 0.3 and 0.5 when the DST is applied, indicating that the DST is successful in destroying the hidden data within the video stream. It is also observable through figures 9.1b and 9.2b that the quality of the DST attacked cover media is acceptable for all simulated Spring



(a) BER vs Pinch Width

(b) PSNR vs Pinch Width

Figure 9.2: RST Attack

transform widths, where the PSNR is approximately 30dB or more. As a result, we can conclude that this simulation proves the Fundamental DST attack is successful in attacking the motion vector and RST-Resilient algorithms.

9.2 Multi-dimensional DST Attack

The next simulation uses the multi-dimensional DST implementation to attack 2D and 3D video steganography schemes. The simulation demonstrates how the DST may be applied in different attack vectors to attempt to disrupt steganographic schemes that may be implemented in multiple dimensions. The algorithm used for the simulation is a direct implementation of the scheme proposed in figure 6.1, where each frame is encoded using image-based steganography from [26] and the video's motion is encoded using motion vector steganography from [32].

9.2.1 2D Video DST Attack

Table 9.1 shows the BER for the 2D steganographic scheme under the multi-dimensional DST attack. As evident, the BER of the 2D steganographic scheme was increased dramatically with the 2D and Time DST attack. The table indicates that the 2D DST attack was much more successful in destroying the steganographic media than was the Time DST attack, however, even for a modest 2D DST attack the BER increases to approximately 0.5 which is shown in highlight.

		2D Chop (pixels)					
		0	10	20	30	40	50
Time Chop (frames)	0	0.0003	0.1020	0.5351	0.5195	0.5185	0.4611
	5	0.0006	0.1033	0.5364	0.5214	0.5077	0.4531
	10	0.0010	0.1043	0.5418	0.5236	0.5086	0.4557
	15	0.0010	0.1052	0.5402	0.5293	0.5140	0.4534
	20	0.0010	0.1043	0.5450	0.5332	0.5057	0.4544
	25	0.0013	0.1087	0.5421	0.5360	0.5061	0.4585

Table 9.1: 2D Steganography BER

9.2.2 Time (3D) DST BER

Next, the 3D DST Table 9.2 shows the BER for the motion vector scheme under the multi-dimensional DST attack. It can be seen that the BER of the 3D steganographic scheme was increased significantly under either the 2D or Time DST attack. Unlike the 2D steganographic scheme, both the 2D and Time DST attacks were equally successful in combating the motion vector steganography, as evidenced by the fact both schemes successfully increase the BER to 0.5 dB.

9.2.3 Cover Media Quality

Table 9.3 shows the PSNR for the video sequence under the multi-dimensional DST attack. The results indicate that the PSNR was acceptable for all tested DST

		2D Chop (pixels)					
		0	10	20	30	40	50
Time Chop (frames)	0	0.0000	0.0000	0.3891	0.4713	0.5161	0.5146
	5	0.5197	0.5257	0.5106	0.5338	0.4970	0.5015
	10	0.5423	0.5302	0.5474	0.4945	0.5121	0.5287
	15	0.4990	0.5318	0.5484	0.5378	0.5484	0.5237
	20	0.5302	0.5635	0.5181	0.5297	0.5297	0.5156
	25	0.5514	0.5559	0.5413	0.5398	0.4950	0.5186

Table 9.2: Motion Vector Steganography BER

transforms, where the PSNR always remains close to 30dB. The optimal transform widths for these schemes were indicated by the shaded cell in each of the three tables, where the 2D DST Chop was 20 pixels and the Time (3D) DST Chop was 10 frames.

		2D Chop (pixels)					
		0	10	20	30	40	50
Time Chop (frames)	0	∞	30.887	29.686	29.339	29.144	29.018
	5	34.250	30.136	29.377	29.137	28.994	28.898
	10	32.238	29.765	29.223	29.036	28.913	28.828
	15	31.470	29.594	29.141	28.979	28.868	28.792
	20	31.090	29.496	29.092	28.943	28.838	28.769
	25	30.864	29.435	29.059	28.919	28.817	28.752

Table 9.3: Motion Vector Steganography PSNR (db)

9.3 Domain-based DST Attack

Next, we simulated the domain-based DST implementation by attacking the frequency domain of an image encoded using image-based steganography, which we call the Frequency Discrete Spring Transform (FDST) attack. We encoded a multitude of images using a simple spread-spectrum steganographic technique [26] where the images were selected from the USC SIPI [82], and Kodak lossless image suite databases [83]. The spread-spectrum technique encodes information within

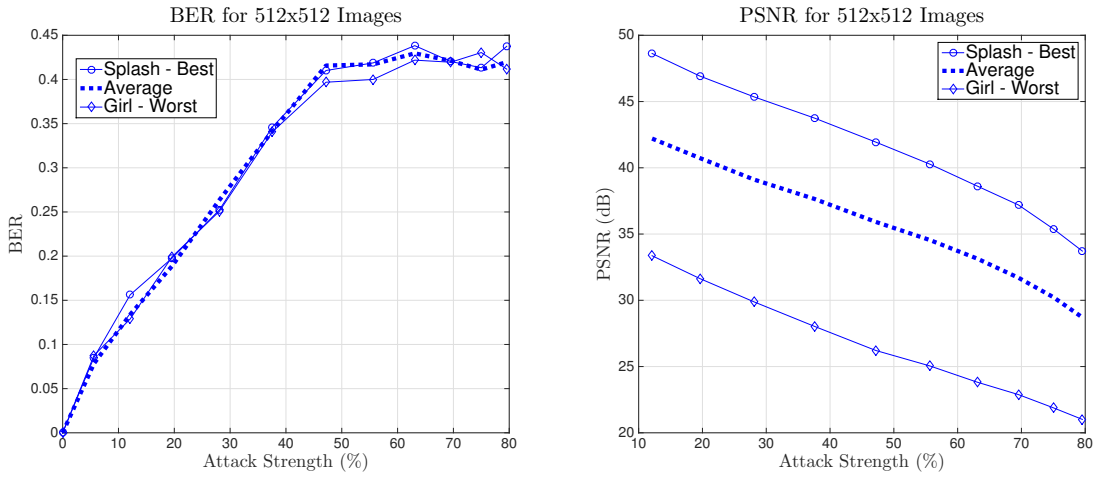
the mid-range frequency components of the frequency domain of the cover media, and is a typical example of frequency domain steganography. The important parameters for this attack are related to the selection and partitioning of the mid-range frequency components. For this implementation, we chose to sweep the attack against the percentage of the attacked versus non-attacked frequency components of the image's frequency domain representation, which we refer to as the attack strength. This is to avoid severely distorting the image while also utilizing the knowledge that most steganographic schemes which encode information in the frequency domain utilize the mid-range frequency components.

Our simulation results indicate that increasing the size of the mid-range selection increases the BER of the encoded information while decreasing the PSNR of the image. Fig 9.3a demonstrates the BER of the 512 by 512 pixel images under FDST attack. The figure demonstrates that the DST is effective in destroying the hidden stego-data of the cover media, as the BER increases steadily to 0.5 with sufficient attack strength.

In order for the attack against the 512 by 512 pixel images to be considered successful, the image must maintain an acceptable quality, where a PSNR of 30dB is typically considered acceptable. Fig 9.3b shows the PSNR of the images under attack. Although the quality drops according to the attack strength, the figure indicates that the images maintain an acceptable quality and proves the attack is not too damaging.

To further strengthen these results, we performed the attack against another set of test images of size 768 by 512 pixels. Again, the results indicate the attack was successful at removing the stego-data from the cover media.

Fig 9.4a shows the BER for the set of 768 by 512 pixel images under FDST attack. The BER increases dramatically at approximately 30% attack strength to



(a) BER vs Attack Strength

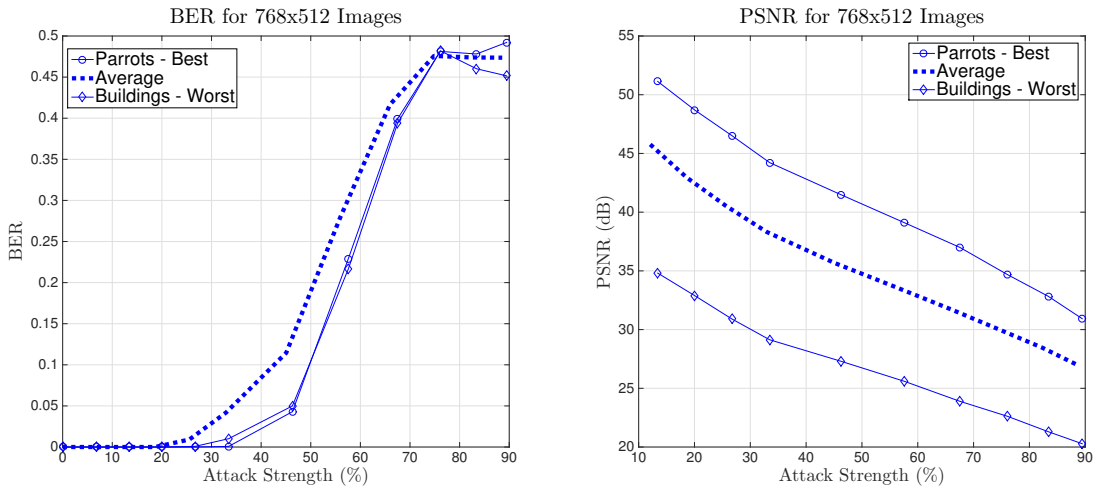
(b) PSNR vs Attack Strength

Figure 9.3: 512x512 Image Attack

approximately 0.5, indicating that the stego-data is completely destroyed within the cover media.

The quality of the 768 by 512 pixel images was also assessed in order to prove that the attack was not too damaging to the images. As seen in figure 9.4b, the images maintain a strong PSNR of approximately 30dB, which means that the quality is preserved for the attack.

It follows that the FDST attack is effective at destroying the steganographic information within all of the attacked images. We found that certain images were more affected by the attack than others, for example, the images Girl and Buildings both produced a much worse PSNR than the average PSNR of the other images. The properties of these images are such that they were more severely affected by distortions in the mid-range frequency components than the other test images, likely due to the fact that there are a lot of smaller details within these images that would make them more susceptible to frequency domain distortion than others. However, despite this fact, the BER for these two images remained static as



(a) BER vs Attack Strength

(b) PSNR vs Attack Strength

Figure 9.4: 768x512 Image Attack

compared with the average BER, which means an attacker could simply choose the PSNR they would like to maintain in order to maximize the BER of the image. It is important to note that a steganographer would also face similar difficulties for images which are more susceptible to mid-range frequency distortion as they would not be able to encode stego-data as strongly.

9.4 Multi-Vector DST Attack

Our Multi-Vector DST attack was broken into two distinct sub-attacks to demonstrate important implementations. Firstly, an entirely PFO-based attack was implemented in order to demonstrate how to maximize the effectiveness of the attack. This PFO attack operates on a feedback loop to maximize the performance and quality of the attacked media. Using the results of this attack, one can begin to select good approximations of various components of the MV-DST attack and generalize how properties such as reactivity, elasticity, and continuity correlate to performance and quality of attacked media. Using this as a basis, the MV-DST

can be implemented to attack multiple simultaneous vectors within a cover media using approximations found in the PFO-based attack.

9.4.1 Perceptually Faithful Only Attack

The PFO DST attack was implemented against a collection of images from the USCI [82] image database. Although the attack may easily be extrapolated to other media types, images were chosen due to the plethora and diverse set of image-based steganographic techniques.

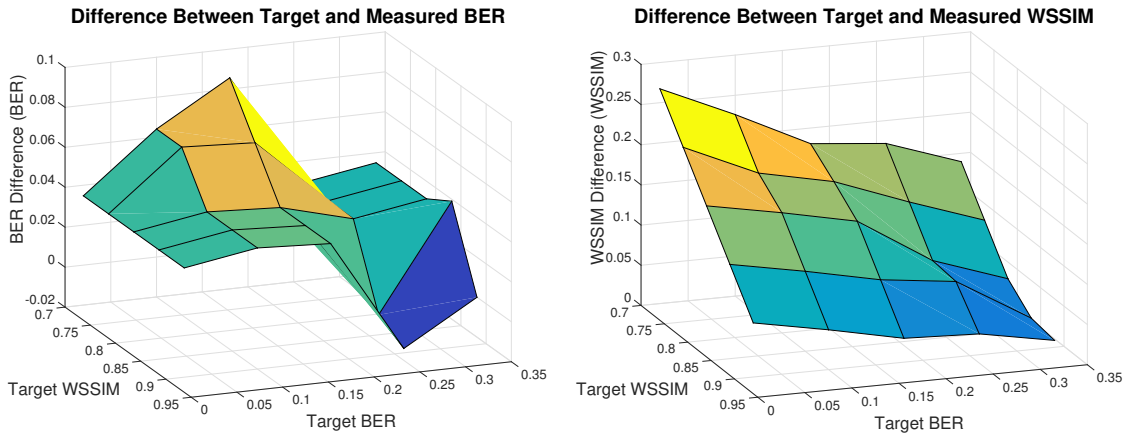
Three distinct steganographic algorithms were used to encode data within this image set which use Spread-Spectrum (SS), Single Value Decomposition (SVD), and Rotation Scaling Translation-resilient (RST) embedding techniques. The SS-based technique is a very typical type of image steganography, and the algorithm used for this attack [84] embeds spreading sequences in the DCT domain of an image. The SVD-based technique is more sophisticated than the SS-based technique and uses the Discrete Wavelet Transform (DWT) to embed components in the HH block of the image's DWT domain. The specific algorithm used for the SVD implementation is described in [85]. Lastly, the RST-based technique uses normalization domains to embed components in a domain which is guaranteed to remain static under RST attacks. RST methods represent some of the most sophisticated embedding techniques, and are some of the only techniques which have been proven to actively resist active steganographic attacks. The algorithm used for this RST embedding technique is based on a normalization-based technique described in [81].

The goal of this attack was to match a set of performance and quality metrics as closely as possible. In a typical attack scenario, an attacker would attempt to maximize the errors in the steganographic data, while minimizing the damage done to the media. The purpose of matching target performance and quality metrics

(rather than maximizing performance and quality) is to show the adaptability of the PFO DST attack and how it can be easily tuned to fit specific attack parameters. The algorithm described in figure 8.1 was utilized in order to estimate parameters that matched the target performance and quality. The attacks were measured by computing the difference between the target and measured attack parameters. A difference of 0 is ideal and indicates that the attack perfectly matched the attack parameters.

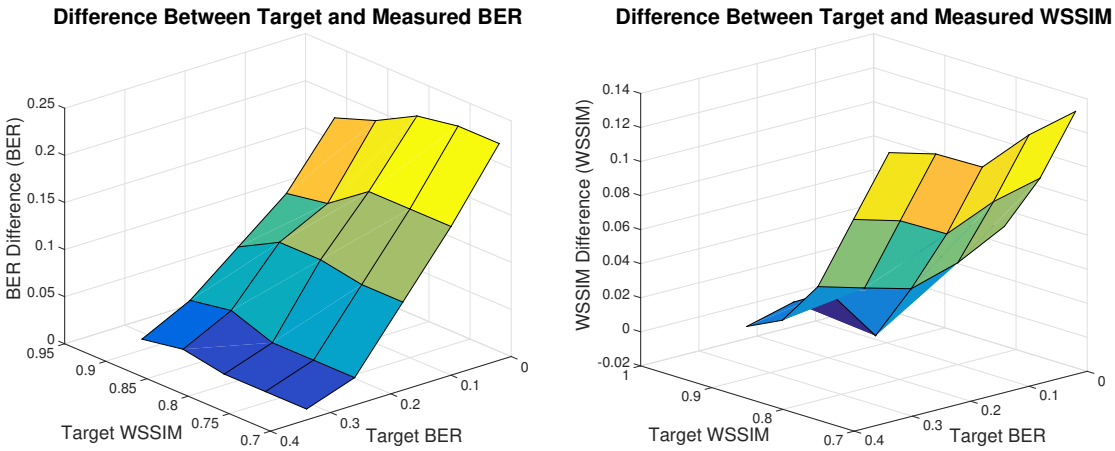
The results of the performance for the three steganographic algorithms are shown in figures 9.5a, 9.6a, and 9.7a. The results indicate the PFO DST attack was capable of matching the target performance parameters, where nearly all attack parameters were met or exceeded. As evident, certain attack parameters produced a larger difference score than others, however, all performance differences were greater than 0, which indicates that all attack parameters exceeded the target performance metric. Note that certain combinations of attack parameters are impractical or impossible to achieve. For example, consider a target performance metric 0.05 (BER) and target quality metric of 0.7 (WSSIM). This combination means that the quality of the media is rather low, while the steganographic performance is high. This combination is difficult to achieve because it means the media has been changed a lot (perceptually and numerically), but the performance of the algorithm has remained high. For situations like this, we see that the performance difference of the attack was greatly exceeded, which makes sense considering it is an impractical combination to achieve.

The results of the quality for the three steganographic algorithms is shown in figures 9.5b, 9.6b, and 9.7b. The results indicate that the PFO DST attack was able to match the quality parameters of the attack. Again, all attack parameters produced difference scores greater than 0. This is again explained by certain



(a) Performance (b) Quality

Figure 9.5: SS Attack

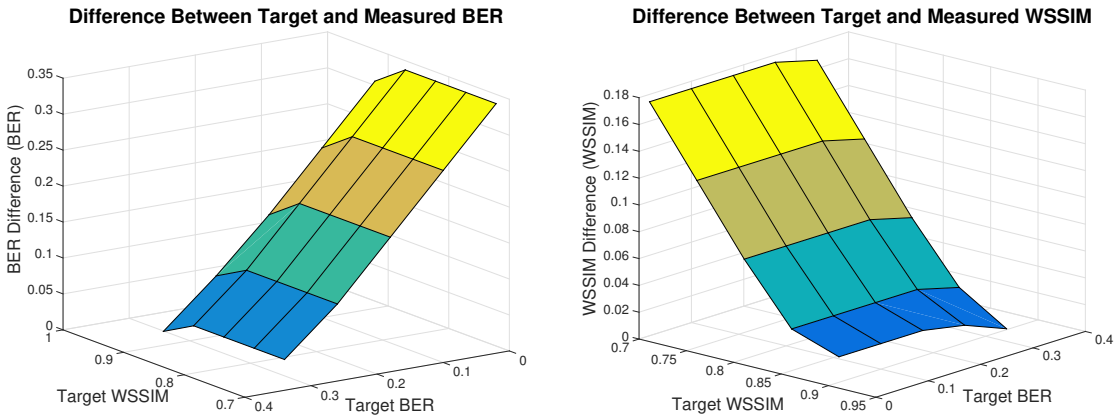


(a) Performance (b) Quality

Figure 9.6: SVD Attack

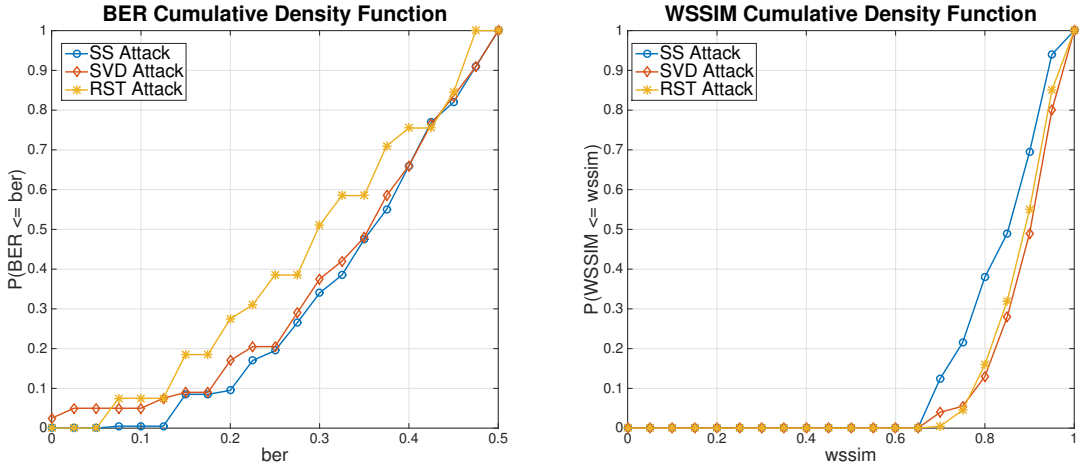
unrealistic attack parameter combinations. Despite some of these non-zero scores, they are expected and do not detract from the performance results of the attack.

The Cumulative Density Function (CDF) for the performance and quality of these three attacks is shown in figure 9.8. Ideally, if the attack perfectly matched all target attack parameters, the trend of the CDF would be a straight line. The CDF trend shows the PFO DST attack was capable of matching the attack parameters in



(a) Performance (b) Quality

Figure 9.7: RST Attack



(a) Performance CDF (b) Quality CDF

Figure 9.8: Simulation Results - CDF

a reasonable manner, as the CDF is zero for reasonable target metrics.

The importance of these results is not that the steganographic algorithms were defeated (although this is certainly important) but that the attack vectors (BER and WSSIM) were directly specified and achieved using the PFO DST attack. In this manner, the PFO DST can be used to directly compute a tradeoff between performance and quality to select an attack which is optimal for a particular target

stego-media. When using approximations of the quality and performance for a stego-media under attack, the PFO DST attack can be realistically achieved in a real-world environment to ensure that performance and quality are maintained for a particular attack. This approach to steganographic attacks is fundamentally unique from other active approaches, and can be considered the optimal method of removing stego-data from a cover media.

9.4.2 Multi-Vector Attack

Next, the MV-DST was simulated against various types of image-based steganographic algorithms. A set of 37 images from the SCISC [82] database were encoded using three distinct steganographic algorithms which embed components using Discrete Cosine Transform (DCT), Rotation Scaling Translation (RST), and Singular Value Decomposition (SVD)-based embedding techniques and attacked using the MV-DST. These techniques represent various degrees of sophistication, where the DCT embedding algorithm would be considered the most simplistic and fragile, and the RST and SVD the most complex and robust. The MV-DST was limited to image-based algorithms, as the most prolific and diverse algorithms are also image-based, but the approach could easily implemented to non-2-dimensional and non-image derived media as well.

To ease the computational complexity of the MV-DST, the Spring-Mesh Φ was only implemented in the Γ^{th} (positional) dimension, meaning the amplitudinal values of the media were left unaltered. Φ_Γ may be observed in figure 9.9, and remained static for all vectors.

The first attack was against a spread-spectrum based technique that embeds steganographic data using a spreading sequence in the DCT coefficients of an image [84]. This technique, while simple, is among a class of algorithms that is

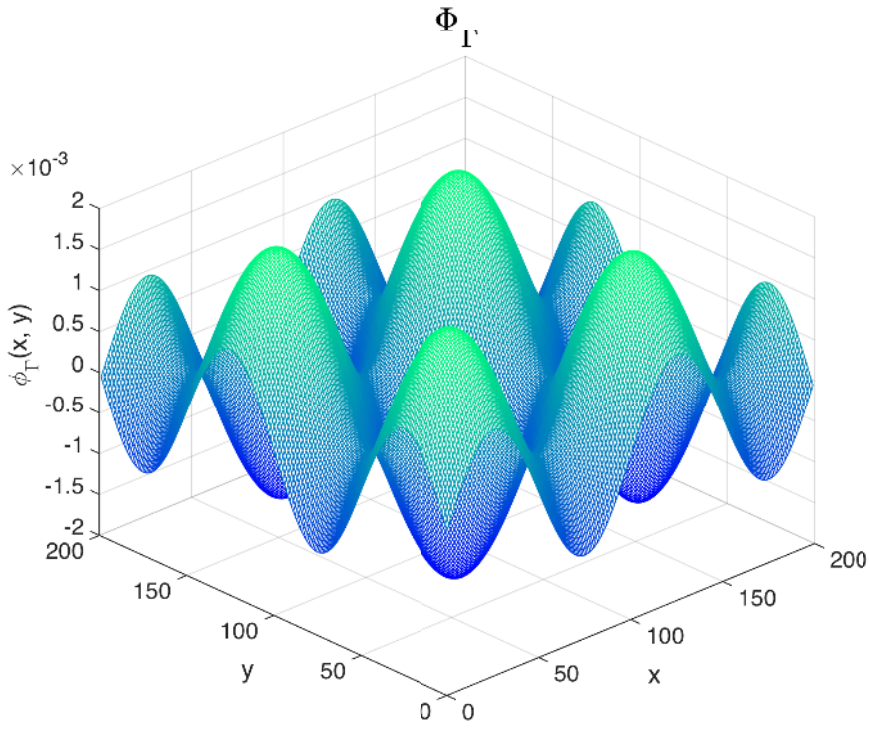
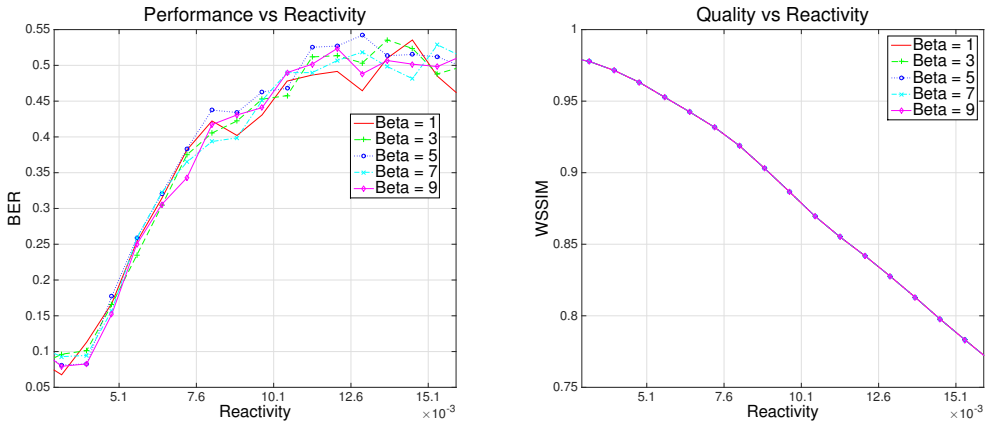


Figure 9.9: Φ_Γ - Spring Mesh for Attack

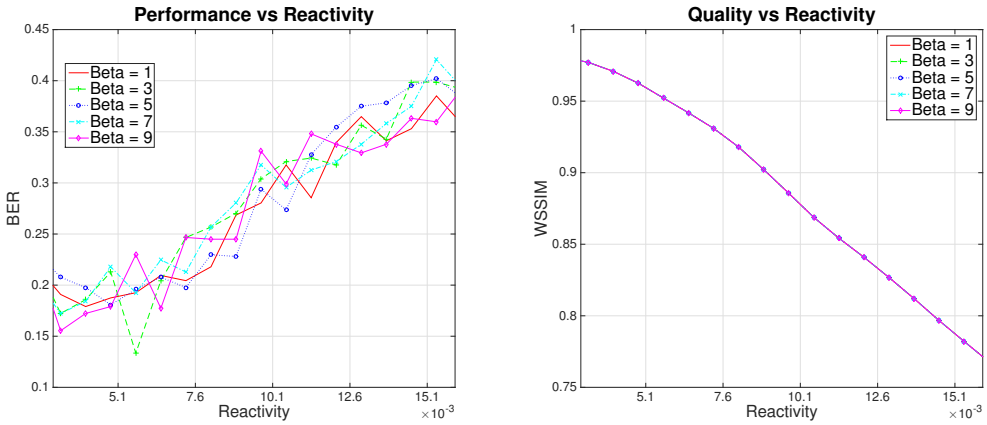
extremely common. In fact, many prolific image-based steganographic algorithms such as F5 [86], YASS [87], and OutGuess [68] are based on embedding within a frequency-derived domain of an image. Figures 9.10a and 9.10b show the results of the MV-DST, where the performance and quality of the media after the attack was analyzed. The performance trend shows that as the reactivity ρ of Φ increases the performance (as measured using Bit Error Rate (BER)) decreases. A BER of 0.5 indicates that the steganographic data within the media is completely unrecoverable. This indicates that as ρ is increased, the steganographic data contained within is destroyed. Likewise, the quality of the media was observed using the Weighted Structural Similarity Index (WSSIM). The quality trend shows that as ρ is increased, the quality decreases as well, but maintains an acceptable level of over 0.75 for all observed ρ.



(a) Performance vs Reactivity (b) Quality vs Reactivity

Figure 9.10: DCT MV-DST Attack

The second attack was targeted at a Singular Value Decomposition (SVD)-based technique, where the Discrete Wavelet Transform (DVT) is used to embed steganographic components in the cover media. SVD techniques typically operate by performing SVD on a target block of a cover media, typically in a domain such as the DWT, and embedding bits. The algorithm for this approach described in [85] embeds bits using SVD of the HH block of the DWT domain. SVD-based techniques [85, 88, 89] are typically considered more robust than simple DCT and frequency-based techniques, at the cost of added complexity and reduced capacity. The results for this attack are shown in figure 9.11, where the performance and quality of the media under MV-DST are assessed. Figure 9.11a compares the performance (BER) versus the reactivity, where the results indicate that the performance decreases (BER approaches 0.5) as ρ increases. Likewise, the quality of the encoded media was also measured, where the quality decreases but maintains acceptable levels (WSSIM is greater than 0.75) as ρ increases. The results show that the steganographic data was destroyed while maintaining acceptable quality levels for all simulated ρ.

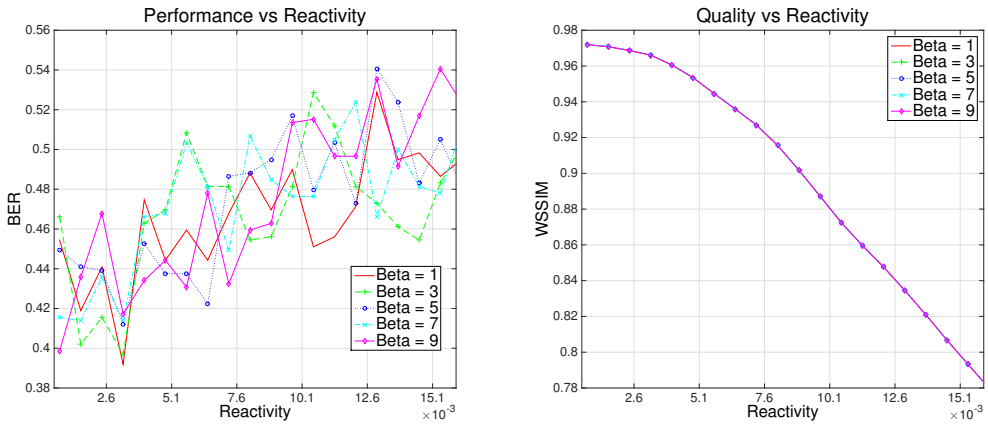


(a) Performance vs Reactivity (b) Quality vs Reactivity

Figure 9.11: SVD MV-DST Attack

The results for the third attack were against a Rotation Scaling Translation (RST) resilient steganographic technique. RST techniques are capable of resisting attack from basic image manipulation techniques that include simple affine transformations, as well as scaling and rotational operations [36, 81, 90]. These techniques typically operate by embedding information within a domain of an image that is guaranteed to remain static under RST operations. Such techniques are important as they are among a class of algorithms that are capable of resisting active steganographic attacks. The technique used is based on normalization domains as described in [81], and embeds steganographic components in this RST-resilient normalization domain. The results of this attack are shown in figures 9.12a and 9.12b, where the performance and quality were analyzed against the reactivity ρ. Figure 9.12a shows the performance trend, where it can be observed that the performance decreases (BER increases) as ρ increases. The BER approaches 0.5 as ρ increases, which indicates that the steganographic data within the media is non-recoverable. Likewise, the performance was analyzed in figure 9.12b, where the quality of the media versus ρ is shown. As ρ increases, the quality of the media

(as measured in WSSIM) decreases, but remains high (above 0.75) for the range of ρ.



(a) Performance vs Reactivity (b) Quality vs Reactivity

Figure 9.12: RST MV-DST Attack

The results of the MV-DST attacks show that the MV-DST was capable of defeating the steganographic algorithms while maintaining a high perceptual quality for all observed reactivity (ρ). An interesting observation is that the choice of β did not significantly impact the quality of the media, where the trends for each β were identical, yet the BER trends were impacted slightly. Since the choice of β was relatively small (between 1 and 9), this was probably not a significant impact on the quality and performance of the attacks. However, a choice of a larger β could perhaps be used to further tune the target metrics of the MV-DST.

Interestingly, the RST-resilient technique, which was thought to have been the most robust to active methods, had the worst performance of the three. This could possibly be attributed to the fact the MV-DST does not have as much redundancy as the others (it embeds fewer bits per message bit), however it also may indicate that the MV-DST is distinct from simple RST image manipulation methods.

Chapter 10

Conclusion

The Discrete Spring Transform is an active steganographic attack that exploits weaknesses in modern steganographic algorithms. The attack operates by disrupting numeric components in a cover media while minimizing degradation to perceptual quality. My research efforts have shown that steganographic algorithms are easily disrupted by changes in their cover media's numeric representation, thus, by exploiting this shortcoming, our DST is able to render a cover media unsuitable for carrying stego-data. Since the DST is an active attack, it can be applied to any digital media and tuned using various properties that dictate the strength and directionality of the attack. My numerous attack simulations have shown that the DST is capable of defeating numerous types of steganographic algorithms in a variety of cover media. These results correspondingly indicate that the DST can render the effective BER of most steganographic algorithms to approximately 0.5 (indicating complete randomness in the received of stego-data) while maintaining strong perceptual quality of the cover media. The DST has even defeated steganographic algorithms that are resistant to attack, further demonstrating the effectiveness of the technique and how it is distinct from simple signal processing operations.

The DST is unique from the vast majority of steganographic attacks in that it is

active rather than passive. In modern, widely distributed communication networks, such as the internet, passive attacks begin to breakdown since it is impractical to globally assess and analyze suspect media on a case-by-case basis. An active attack does not have such a restriction, and can be applied without prior knowledge or study of suspected algorithms. The DST is distinguished from the small set of active steganographic attacks in that it is the first attack that exploits perceptual identity in media as a basis for attack. In other words, the DST exploits the fact that two digital media can be rendered in perceptually identical, yet numerically distinct ways. The DST thus renders a digital cover media unsuitable for carrying steganographic data, and is unique in its perception-preserving approach that it utilizes.

I believe that the Discrete Spring Transform can address shortcomings in modern steganographic attacks. The DST employs a communications-theoretic approach to defeating steganography that can be applied to virtually any digital media containing virtually any type of stego-data. As a result, my research efforts have shown that the DST is an effective steganographic attack that is capable of defeating steganography in a variety of digital mediums in a highly efficient and adaptable manner.

Bibliography

[1] A. Sharp, Qilin Qi, Yaoqing Yang, Dongming Peng, and H. Sharif. A novel active warden steganographic attack for next-generation steganography. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1138–1143, July 2013. 1, 1, 6.3.1, 7.1.1, 7.2, 8.2, 8.2.3

[2] A. Sharp, Qilin Qi, Yaoqing Yang, Dongming Peng, and H. Sharif. A video steganography attack using multi-dimensional discrete spring transform. In *Signal and Image Processing Applications (ICSIPA), 2013 IEEE International Conference on*, pages 182–186, Oct 2013. 2, 1, 8.2

[3] Qilin Qi, A. Sharp, Dongming Peng, Yaoqing Yang, and H. Sharif. An active audio steganography attacking method using discrete spring transform. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 3456–3460, Sept 2013. 3, 1, 8.2

[4] A. Sharp, Qilin Qi, Yaoqing Yang, Dongming Peng, and H. Sharif. Frequency domain discrete spring transform: A novel frequency domain steganographic attack. In *Communication Systems, Networks Digital Signal Processing (CSNDSP), 2014 9th International Symposium on*, pages 972–976, July 2014. 4, 1, 8.2

[5] Qilin Qi, A. Sharp, Yaoqing Yang, Dongming Peng, and H. Sharif. Steganography attack based on discrete spring transform and image geometrization.

In *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*, pages 554–558, Aug 2014. 5, 1, 8.2

[6] Aaron Sharp and Dongming Peng. The multi-vector discrete spring transform. *Journal of Information Security and Applications*, 2017. Publication Pending. 6, 1

[7] Aaron Sharp and Dongming Peng. An active steganographic attack approach based on perception-preserving discrete spring transform. *Journal of Information Security and Applications*, 2017. Publication Pending. 7, 1

[8] Bin Li, Junhui He, Jiwu Huang, and Yun Qing Shi. A survey on image steganography and image steganalysis. *Journal of Information Hiding and Multimedia Signal Processing*, 2011. 1, 2, 3, 3.1, 3.1.1, 3.2

[9] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding – a survey, 1999. 1, 2, 3, 3.1, 3.1.1

[10] Dong Zheng, Yan Liu, and Jiying Zhao. A survey of rst invariant image watermarking algorithms. In *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, pages 2086–2089, 2006. 1, 3.1.3

[11] Siwei Lyu and Hany Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *In 5th International Workshop on Information Hiding*, pages 340–354. Springer-Verlag, 2002. 1, 2, 3.2.2

[12] Siwei Lyu and Hany Farid. Steganalysis using color wavelet statistics and one-class support vector machines. In *In SPIE Symposium on Electronic Imaging*, pages 35–45, 2004. 1, 2, 3.2.2

[13] Gokhan Gul and Fatih Kurugollu. A new methodology in steganalysis: Breaking highly undetectable steganograpy (hugo). In Tom Filler, Tom Pevn,

Scott Craver, and Andrew Ker, editors, *Information Hiding*, volume 6958 of *Lecture Notes in Computer Science*, pages 71–84. Springer Berlin Heidelberg, 2011. 1, 2, 3.2.2

[14] Tomas Pevny and Jessica Fridrich. Merging markov and dct features for multi-class jpeg steganalysis, 2007. 1, 2, 3.2.2

[15] Fabien A P Petitcolas. Watermarking schemes evaluation. *Signal Processing Magazine, IEEE*, 17(5):58–64, Sep 2000. 1, 3.3, 3.4.3

[16] FabienA.P. Petitcolas, RossJ. Anderson, and MarkusG. Kuhn. Attacks on copyright marking systems. In *Information Hiding*, volume 1525 of *Lecture Notes in Computer Science*, pages 218–238. Springer Berlin Heidelberg, 1998. 1, 3.3, 3.4.3

[17] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, April 2004. 1, 4.3.1.2

[18] G.W. Mooney. *De vita Caesarum English & Latin 1930*. Longman, 1930. 2

[19] F.H. Hinsley and A. Stripp. *Codebreakers: The Inside Story of Bletchley Park*. Oxford paperbacks. Oxford University Press, 2001. 2

[20] W. Stallings. *Cryptography and Network Security: Principles and Practice*. The William Stallings books on computer and data communications technology. Prentice Hall, 1999. 2

[21] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *CRYPTO*, pages 51–67, 1983. 2, 3.3

[22] Paul Cruickshank Nic Robertson and Tim Lister. Documents reveal al qaeda's plans for seizing cruise ships, carnage in europe, May 2012. 2

[23] Ten alleged secret agents arrested in the united states, June 2010. 2

[24] Sendatsu. Looking inside your screenshots, September 2012. 2

[25] Christian Rey and Jean-Luc Dugelay. A survey of watermarking algorithms for image authentication. *EURASIP Journal on applied Signal Processing Volume 2002 N6 - June 2002, special issue on image analysis for multimedia interactive services*, 06 2002. 3

[26] Lisa Marvel, Charles Boncelet, and Charles Retter. Spread spectrum image steganography. *IEEE Transactions on Image Processing*, 8:1075–1083, 1999. 3, 3.1.2, 9.2, 9.3

[27] N. Cvejic and T. Seppanen. Increasing the capacity of lsb-based audio steganography. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 336–338, Dec 2002. 3.1.1

[28] N. Cvejic and T. Seppanen. Increasing robustness of lsb audio steganography using a novel embedding method. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 533–537 Vol.2, April 2004. 3.1.1

[29] R. Chandramouli and N. Memon. Analysis of lsb based image steganography techniques. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 1019–1022 vol.3, 2001. 3.1.1

[30] Weiqi Luo, Fangjun Huang, and Jiwu Huang. Edge adaptive image steganography based on lsb matching revisited. *Information Forensics and Security, IEEE Transactions on*, 5(2):201–214, June 2010. 3.1.1

[31] ShengDun Hu and U. KinTak. A novel video steganography based on non-uniform rectangular partition. In *Computational Science and Engineering (CSE), 2011 IEEE 14th International Conference on*, pages 57–61, 2011. 3.1.1

[32] A.T. Sharp, J. Devaney, and A.E. Steiner. Digital video authentication with motion vector watermarking. In *Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on*, pages 1 –4, dec. 2010. 3.1.1, 5.3.1, 5.3.3, 6.1.2, 6.3, 9.1, 9.2

[33] N. Mohaghegh and O. Fatemi. H.264 copyright protection with motion vector watermarking. In *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, pages 1384 –1389, july 2008. 3.1.1, 5.3.1, 6.1.2, 6.3

[34] Andreas Westfeld. F5: a steganographic algorithm: High capacity despite better steganalysis. In *4th International Workshop on Information Hiding*, pages 289–302. Springer-Verlag, 2001. 3.1.2

[35] Jessica Fridrich, Miroslav Goljan, and Dorin Hogea. Attacking the outguess, 2002. 3.1.2

[36] Ching-Yung Lin, Min Wu, J.A. Bloom, Ingemar J. Cox, M.L. Miller, and Yui Man Lui. Rotation, scale, and translation resilient watermarking for images. *Image Processing, IEEE Transactions on*, 10(5):767–782, 2001. 3.1.3, 9.4.2

[37] Qi Song, Guang xi Zhu, and Hang jian Luo. Geometrically robust image watermarking based on image normalization. In *Intelligent Signal Processing*

and Communication Systems, 2005. ISPACS 2005. Proceedings of 2005 International Symposium on, pages 333–336, 2005. 3.1.3

[38] Hwan Il Kang and E.J. Delp. An image normalization based watermarking scheme robust to general affine transformation. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 3, pages 1553–1556 Vol. 3, 2004. 3.1.3

[39] Dong Zheng, Jiying Zhao, and A.E. Saddik. Rst-invariant digital image watermarking based on log-polar mapping and phase correlation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(8):753–765, 2003. 3.1.3

[40] M. Hemahlathaa and C. Chellppan. A feature-based robust digital image watermarking scheme. In *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, pages 1–5, 2012. 3.1.3

[41] M.S. Yasein and P. Agathoklis. An image normalization technique based on geometric properties of image feature points. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 116–121, 2007. 3.1.3

[42] Wei Li, Xiangyang Xue, and Peizhong Lu. Localized audio watermarking technique robust against time-scale modification. *Multimedia, IEEE Transactions on*, 8(1):60–69, Feb 2006. 3.1.3

[43] Ryuki Tachibana, Shuichi Shimizu, Taiga Nakamura, and Seiji Kobayashi. Audio watermarking method robust against time- and frequency-fluctuation, 2001. 3.1.3

[44] M.F. Mansour and A.H. Tewfik. Audio watermarking by time-scale modification. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 3, pages 1353–1356 vol.3, 2001. 3.1.3

[45] M.F. Mansour and A.H. Tewfik. Time-scale invariant audio data embedding. In *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, pages 76–79, Aug 2001. 3.1.3

[46] I. Avcibas, N. Memon, and B. Sankur. Steganalysis using image quality metrics. *Image Processing, IEEE Transactions on*, 12(2):221–229, 2003. 3.2.1

[47] A.D. Ker. Steganalysis of lsb matching in grayscale images. *Signal Processing Letters, IEEE*, 12(6):441–444, 2005. 3.2.1

[48] JiFeng Huang and Qingju Jiao. A steganalysis method based on quantization attack. In *Image and Signal Processing, 2008. CISP '08. Congress on*, volume 5, pages 640–644, 2008. 3.2.1

[49] Siwei Lyu and H. Farid. Steganalysis using higher-order image statistics. *Information Forensics and Security, IEEE Transactions on*, 1(1):111–119, March 2006. 3.2.1

[50] Yun Q. Shi, Chunhua Chen, and Wen Chen. A markov process based approach to effective attacking jpeg steganography. In *Proceedings of the 8th International Conference on Information Hiding*, IH'06, pages 249–264, Berlin, Heidelberg, 2007. Springer-Verlag. 3.2.1

[51] Chunhua Chen and Y.Q. Shi. Jpeg image steganalysis utilizing both intrablock and interblock correlations. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 3029–3032, May 2008. 3.2.1

[52] J. Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *Information Forensics and Security, IEEE Transactions on*, 7(2):432–444, April 2012. 3.2.2

[53] Jessica Fridrich, Jan Kodovsk, Vojtech Holub, and Miroslav Goljan. Steganalysis of content-adaptive steganography in spatial domain. In Toms Filler, Toms Pevn, Scott Craver, and Andrew D. Ker, editors, *Information Hiding*, volume 6958 of *Lecture Notes in Computer Science*, pages 102–117. Springer, 2011. 3.2.2

[54] A.M. Eskicioglu and P.S. Fisher. Image quality measures and their performance. *Communications, IEEE Transactions on*, 43(12):2959–2965, Dec 1995. 3.3

[55] B. Girod. Psychovisual aspects of image processing: What's wrong with mean squared error? In *Multidimensional Signal Processing, 1991., Proceedings of the Seventh Workshop on*, pages P.2–P.2, Sep 1991. 3.3

[56] Zhou Wang and A.C. Bovik. A universal image quality index. *Signal Processing Letters, IEEE*, 9(3):81–84, March 2002. 3.3

[57] Chao Gan, Xiangyang Wang, Mengyao Zhu, and Xiaoqing Yu. Audio quality evaluation using frequency structural similarity measure. In *Wireless Mobile and Computing (CCWMC 2011), IET International Communication Conference on*, pages 299–303, Nov 2011. 3.3

[58] Daniel L. Currie, Nab Little Creek, and Cynthia E. Irvine. Surmounting the effects of lossy compression on steganography. In *In National Information System Security Conference*, pages 194–201, 1996. 3.3

[59] Anthony Whitehead. Towards eliminating steganographic communication. In *PST*, 2005. 3.3

[60] R. Krenn. Steganography and steganalysis. 3.3

[61] Neil F. Johnson and Sushil Jajodia. Steganalysis of images created using current steganography software. In David Aucsmith, editor, *Information Hiding*, volume 1525 of *Lecture Notes in Computer Science*, pages 273–289. Springer, 1998. 3.3

[62] Christian Cachin. Digital steganography. In *Encyclopedia of Cryptography and Security (2nd Ed.)*, pages 348–352. 2011. 3.3

[63] C.B. Smith and S.S. Agaian. Denoising and the active warden. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 3317–3322, 2007. 3.3

[64] P.L. Shrestha, M. Hempel, Tao Ma, Dongming Peng, and H. Sharif. A general attack method for steganography removal using pseudo-cfa re-interpolation. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 454–459, 2011. 3.3

[65] P.L. Shrestha, M. Hempel, Tao Ma, Dongming Peng, and H. Sharif. Watermark removal using pseudorandom desynchronization by selective pixel elimination. In *Communications (ICC), 2012 IEEE International Conference on*, pages 1016–1020, 2012. 3.3

[66] F. Rezaei, M. Hempel, P.L. Shrestha, Tao Ma, Dongming Peng, and H. Sharif. A quality-preserving hidden information removal approach for digital images. In *Communications (ICC), 2012 IEEE International Conference on*, pages 1021–1025, 2012. 3.3

[67] Gina Fisk, Mike Fisk, Christos Papadopoulos, and Joshua Neil. Eliminating steganography in internet traffic with active wardens. In FabienA.P. Petitcolas, editor, *Information Hiding*, volume 2578 of *Lecture Notes in Computer Science*, pages 18–35. Springer Berlin Heidelberg, 2003. 3.3

[68] Eric Cole. *Hiding in Plain Sight: Steganography and the Art of Covert Communication*. John Wiley & Sons, Inc., New York, NY, USA, 1 edition, 2003. 3.4.1, 9.4.2

[69] Niels Provos and Peter Honeyman. Hide and seek: An introduction to steganography. *IEEE Security and Privacy*, 1(3):32–44, 2003. 3.4.1

[70] Ren Rosenbaum and Heidrun Schumann. A steganographic framework for reference colour based encoding and cover image selection. In Gerhard Goos, Juris Hartmanis, Jan Leeuwen, Josef Pieprzyk, Jennifer Seberry, and Eiji Okamoto, editors, *Information Security*, volume 1975 of *Lecture Notes in Computer Science*, pages 30–43. Springer Berlin Heidelberg, 2000. 3.4.2

[71] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800–801, June 2008. 4.3.1.1

[72] Zina Liu, Huaqing Liang, Xinxin Niu, and YixianYang. A robust video watermarking in motion vectors. In *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*, volume 3, pages 2358 – 2361 vol.3, aug.-4 sept. 2004. 5.3.1, 6.1.2, 6.3

[73] A. Ceddillo-Hernandez, M. Nakano-Miyatake, L. Rojas-Cardenas, and H. Perez-Meana. Robust video watermarking using perceptual information and motion vector. In *Circuits and Systems, 2007. NEWCAS 2007. IEEE Northeast Workshop on*, pages 811 –814, aug. 2007. 5.3.1, 6.1.2, 6.3

[74] Yuting Su, Chengqian Zhang, and Chuntian Zhang. A video steganalytic algorithm against motion-vector-based steganography. *Signal Process.*, 91(8):1901–1909, August 2011. 5.3.1

[75] Ping Dong, J.G. Brankov, N.P. Galatsanos, Yongyi Yang, and F. Davoine. Digital watermarking robust to geometric distortions. *Image Processing, IEEE Transactions on*, 14(12):2140 –2150, dec. 2005. 5.3.2, 5.3.3

[76] ShengDun Hu and U. KinTak. A novel video steganography based on non-uniform rectangular partition. In *Computational Science and Engineering (CSE), 2011 IEEE 14th International Conference on*, pages 57 –61, aug. 2011. 6.1.1, 6.3

[77] Bin Liu, Fenlin Liu, Chunfang Yang, and Yifeng Sun. Secure steganography in compressed video bitstreams. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 1382 –1387, march 2008. 6.1.1

[78] K. Raghavendra and K.R. Chetan. A blind and robust watermarking scheme with scrambled watermark for video authentication. In *Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference on*, pages 1 –6, dec. 2009. 6.1.1

[79] Chengqian Zhang, Yuting Su, and Chuntian Zhang. A new video steganalysis algorithm against motion vector steganography. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1 –4, oct. 2008. 6.1.2

[80] Yuting Su, Chengqian Zhang, and Chuntian Zhang. A video steganalytic algorithm against motion-vector-based steganography. *Signal Process.*, 91(8):1901–1909, August 2011. 6.1.2

[81] Ping Dong, J.G. Brankov, N.P. Galatsanos, Yongyi Yang, and F. Davoine. Digital watermarking robust to geometric distortions. *Image Processing, IEEE Transactions on*, 14(12):2140–2150, Dec 2005. 9.1, 9.4.1, 9.4.2

[82] Sipi image database. `http://sipi.usc.edu/database/`. 9.3, 9.4.1, 9.4.2

[83] Rich Franzen. Kodak lossless true color image suite, November 1999. 9.3

[84] Ingemar J. Cox, Joe Kilian, F.T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *Image Processing, IEEE Transactions on*, 6(12):1673–1687, Dec 1997. 9.4.1, 9.4.2

[85] Kuo-Liang Chung, Wei-Ning Yang, Yong-Huai Huang, Shih-Tung Wu, and Yu-Chiao Hsu. On svd-based watermarking algorithm. *Applied Mathematics and Computation*, pages 54–57, 2007. 9.4.1, 9.4.2

[86] Andreas Westfeld. F5-a steganographic algorithm. In *Proceedings of the 4th International Workshop on Information Hiding*, IHW '01, pages 289–302, London, UK, UK, 2001. Springer-Verlag. 9.4.2

[87] Kaushal Solanki, Anindya Sarkar, and B. S. Manjunath. Yass: yet another steganographic scheme that resists blind steganalysis. In *in 9th Int. Workshop on Info. Hiding*, 2007. 9.4.2

[88] Ruizhen Liu and Tieniu Tan. An svd-based watermarking scheme for protecting rightful ownership. *Multimedia, IEEE Transactions on*, 4(1):121–128, Mar 2002. 9.4.2

[89] Emir Ganic and Ahmet M. Eskicioglu. Robust dwt-svd domain image watermarking: Embedding data in all frequencies. In *Proceedings of the 2004 Workshop on Multimedia and Security*, MM&Sec '04, pages 166–174, New York, NY, USA, 2004. ACM. 9.4.2

[90] Dong Zheng, Jiying Zhao, and A.E. Saddik. Rst-invariant digital image watermarking based on log-polar mapping and phase correlation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(8):753–765, Aug 2003. 9.4.2