Table of Contents

DEFINITION **3**

HISTORY **3**

Ancient Times 3

Middle Ages 4

Modern Times 5

Present Day 7

DIGITAL WATERMARKING **7**

DIGITAL WATERMARKING TECHNIQUES **8**

Data-Hiding in Text 9
- Line-shift coding 10
- Word-shift coding 10
- Feature Coding 11
- Other Methods 11

Data-Hiding in Images 12
- Least Significant Bit insertion 13
- Algorithms and Transformations 14

Data-Hiding in Audio 15
- Least Significant Bit Insertion 16
- Phase Coding 17
- Spread Spectrum Coding 17
- Echo Data Hiding 18

Comparison of Techniques 19

WATERMARK ATTACKS **20**

Simple Attacks 20

Detection-Disabling Attacks 21
- Mosaic Attack 21
- Jitter Attack 22

Ambiguity Attacks 22

Removal Attacks 22

Legal Attacks 23

STEGANOGRAPHY SOFTWARE **24**

Software for Testing 24
unZign 24
StirMark 25

Software for Implementation 26
MandelSteg 26
Steg 27
S-Tools 27
Stego 28
Hide and Seek 28
Other Software 29
Software Companies 30

WATERMARKING APPLICATIONS 31

Vatican Library Project 32

Playboy 32

Other Digimarc Ventures 33

Trustworthy Digital Camera 34

FUTURE OF WATERMARKING 34

WORKS CITED 36

Definition

"Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, where the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret message present," describes Dr. Markus Kuhn, computer scientist and Purdue graduate [14].

Steganography can also be explained literally through its Greek meaning, "covered writing". The art of steganography allows messages to be passed covertly without attracting the attention of a third party. These steganographic techniques range from physically hiding the messages, using invisible ink or microdots, to advanced technological methods of watermarking.

History

Ancient Times

The roots of steganography date all the way back to ancient times. The Greeks would often use steganography to send messages in times of war [15]. This technique often helped kings defeat their enemies. In the Histories of Herodotus, many stories illustrated these methods. For example, a message hidden within an animal was one form of steganography. In one story, a messenger disguised as a hunter carried a message to the king by hiding the message in a rabbit's belly. Because the sight of a

hunter carrying its kill through the palace gates aroused no suspicion, the king was able to receive the message.

Messages were also sent through slaves. Kings would often have their slaves' heads shaved and messages tattooed onto their heads. After the hair had grown back, the king would have the slave "personally" deliver the message [15]. No one would be suspecting of these hidden messages unless they knew exactly where to look. Another example of steganography in Ancient Greek times was pricking holes in books above the letters that form the desired message [4]. When the desired recipient obtained the book, that person would knowingly search for the holes to reconstruct the message. To an unknowing reader, the book contained nothing but the author's writing.

The Chinese and Egyptians also had their own methods of steganography in ancient times. The Chinese often wrote messages on thin sheets of silk paper which they rolled into a ball and covered with wax. This ball was then hidden either somewhere on or in their body or swallowed to prevent detection [15]. The Egyptians used illustrations to convey hidden meanings. The Egyptian writing method of hieroglyphics was a common place for secret messages to be kept. When the Egyptians were stopped along delivery, these illustrative writings would not arouse any suspicion from enemies and could be brought to the rightful source [4].

Middle Ages

During the Middle Ages, steganography was further studied and developed. In 1499, Trithemius, a monk, wrote a set of books called Steganographia in which he covered many different techniques of steganography. For example, one form of

steganography that he mentioned included telepathical methods of message passing [15]. Another steganography technique developed in the Middle Ages was the Cardano grille [4]. Developed by Girolamo Cardano, the grille is a sheet of material with randomly cut out rectangles. When the grille is placed over a sheet of paper, the message is written by writing one character or group of characters of the message within the rectangles. Then the grille is removed and the writer fills the remaining space with letters or words to create the message being sent. Once the message is delivered, the receiver places the grille, which is the same as the sender's grille, onto the message and can easily read the underlying message by reading the characters that are in the rectangles.

Early experimentation of invisible ink also began in the Middle Ages. Giovanni Porta wrote several books dealing with natural science [4]. Within these books were recipes for secret ink that could be used to write on human skin and other surfaces. This type of ink was later developed and used in the late 1700's and was the key to secret communications.

Modern Times

Invisible ink was also used heavily in steganography during modern times and is still often used today. Invisible ink was used frequently by spies during WWI and WWII. The alphabetic letters of newspapers or books were often dotted with invisible ink to form messages that were recovered when the newspaper or book was introduced to elements that made the invisible ink darken. Another usage of the ink was to split a piece of paper, use the ink to write the message within the inner surface of the two

pieces, and then rejoin them [4]. Jargon code was also used to deliver secret messages [4]. Jargon code used words in seemingly innocent text that actually stood for other things. For example, in a seemingly innocent letter sent home, family names actually stood for ships or ports. Any jargon code can easily be understood as long as both the receiving and sending parties agree on how to use the code beforehand [4].

Other types of modern day steganography methods include null ciphers and microdots. Null ciphers are messages in which only certain letters are to be used for the underlying message and all other characters are considered to be null or of no value [14]. To use the null cipher correctly, both parties must agree beforehand which characters are of importance. For example, if both parties decided that they are going to use the first letter of every word to form the underlying message, then the sender must put together an innocent text working with those letters. This is often hard to do because the message must make sense to those that come across it so that they do not expect any hidden messages. It would be even harder to form these innocent messages if the agreed characters of importance were every nth character.

Microdots are also another form of steganography used in modern times. The microdot is essentially a photograph of the secret message that is to be delivered. With technological advancements, it is possible to take a picture of the message and shrink it down to a circular photograph of 0.05 inches in diameter [4]. This tiny photograph is then glued onto either a period or a dot of an "i" on another message to be delivered. Only those that know to look for this microdot should be able to detect its presence. Microdots were so successful at one point that they were referred to as, "the enemy's masterpiece of espionage" by FBI Director J. Edgar Hoover [14].

Present Day

With the advancement of present day technology, new steganography methods are produced to complement these new devices. For example, Napster, an Internet site where users can freely download music files, causes many recording artists to use a form of steganography called digital watermarking to protect their content. With the overwhelming use of the Internet in the new millenium, many artists, including musicians, writers, and graphic designers, are becoming concerned with protecting their content from the millions of web surfers. Although certain information can be protected with the use of encryption, digital watermarking is most often used to protect multimedia content either on or off the Internet. Through the use of digital watermarking, illegal copying or misuse of content such as MP3s, DVDs, and even papers, can be reduced.

Digital Watermarking

Digital watermarking is the process of embedding unobtrusive marks or labels, which can be composed of bits or of other data types, into digital content [12]. These watermarks are bound to the source data and are inseparable from the source [12]. This process is considered a form of steganography because the embedded mark is invisible to the human eye when done properly. Only the use of computerized systems can detect the presence of a correctly embedded digital watermark. Later, if necessary, these watermarks can be extracted, without damaging its source, for proof or verification.

Watermarks can serve many purposes. The most prominent applications of digital watermarks are as follows [3][12]:

- **Evidence of ownership:** indicates ownership when the owner's watermark is detected from an illegal copy
- **Fingerprinting:** indicates the identity of an illegal distributor of content or the source of misused content
- **Tracing and Infringement Detection:** can serve as a form of indicators that can trigger protection or royalty collection mechanisms
- **Copy Control:** allows specification of certain features, for example, a file may be used but not copied
- **Labeling and Metadata Insertion:** allows additional information regarding the content or other descriptors to be embedded
- **Authentication:** ensures that the content is authentic and has not been altered since the watermark was inserted

When using digital watermarks for the above purposes, people who feel weary about making their valuable content available on the Internet become less reluctant. By using watermarks to protect the different types of data, content owners can feel more protected.

Digital Watermarking Techniques

There are many different ways to embed a watermark or hide data within other data. To better understand the process of data hiding, certain terms are used when describing the different processes. The data to be hidden will be referred to as the embedded data, the host data is the file in which the data is to be embedded, and the file in which the data has been hidden will be referred to as the cover data.

Watermarkings and other hidden data are usually hidden within text, image, or audio files. This can be done in many different ways. Regardless of the method used to hide this data, certain restrictions must be kept in mind in order to produce a useful

watermark. Bender et al. established guidelines in order to ensure the efficiency and accuracy of the watermarks. These guidelines are as follows [2]:

- The host data should not be degraded and the embedded data should be minimally perceptible.
- The embedded data should be directly encoded into the media so that the data remains intact across varying data file formats.
- The embedded data should be immune to modifications.
- Assymmetrical coding of the embedded data is desirable.
- Error correction coding should be used to ensure data integrity.
- The embedded data should be self-clocking or arbitrarily re-entrant so the embedded data can be recovered when only fragments of the cover data are available.

These are only some of the features that programmers keep in mind when deciding which methods to use in hiding their data. Depending on the type of host data they are using, different techniques may be considered. Although almost all of the following methods can be used for all three different types of data (text, image, and audio files), the ones listed below are the ones that will be covered in greater detail. There are also many other methods that can be used to embed watermarks that are not covered in the following sections.

Data-Hiding in Text

When hiding data in plain text files, methods such as line-shift coding, word-shift coding, and feature coding are often considered. When using a text file as host data, the embedded data is usually a codeword that is hidden within the file by altering different textual features. The three methods listed above determine what feature is to

be changed. To encode the codeword, each bit of the codeword is applied by using one of the three methods.

The lack of mathematical complexity makes these methods easy to implement and understand. These methods are the least technical and can be done even by non-programmers. When used in combinations, these methods provide even better protection for the embedded data.

Line-shift coding

Line-shift coding is very easy to perform and is considered the most resistant to degradation due to copying. In line-shift coding, the lines of text are shifted vertically to encode the document [27]. By determining which lines have been shifted, the encoded bits can be discovered. Although this method withstands copying, the human eye and other measurements can easily detect it. It can also be easily defeated through respacing or reformatting of the text.

Word-shift coding

Word-shift coding can also be easily done. In word-shift coding, codewords are coded into a document by shifting the horizontal location of words within lines of text [27]. In doing so, the appearance of natural spacing must be maintained in order not to arouse suspicion. By determining the location where unnatural spacing has occurred, the encoded bits can be revealed.

There are advantages to using word-shift coding instead of line-shift coding. Word-shift coding is less obvious to the unsuspecting reader. Readers are used to reading text that has been justified for a better presentation. However, there are also

ways that word-shift coding can be detected. If an attacker knew the spacing algorithm, the attacker can calculate the differences in spacing and figure out the encoded data. Like line-shift coding, word-shift coding can also be easily defeated through respacing or justification of the text [4].

Feature Coding

Feature coding is another way of embedding data into a text file. In feature coding, certain text features are altered depending on the embedded data. For example, one type of feature coding would be extending the vertical lines of characters such as "l", "d", "b" and "h" [27]. In order for this type of feature coding to work, the text must be altered by randomizing the lengths of the vertical lines before applying this algorithm. The randomness will help the text look less suspicious to its readers.

In order to decode this algorithm, the text, after the randomization, but before the algorithm application, can be compared with the message containing the embedded data to retrieve the encoded bits. This type of feature coding can be easily defeated if the vertical line length is adjusted to a fixed length before the file is opened.

Other Methods

Line-shift, word-shift, and feature coding may all be applied to text. However, there are some other methods that may be less frequently used. Within this category, there are syntactic and semantic methods. Syntactic methods utilize punctuation and contractions to encode data [29]. However, using this method limits the amount of data that can be encoded. Semantic methods involve word manipulation [29]. By using synonyms and giving values to certain words, the encoded data can easily be retrieved.

For example, words that can be used interchangeably such as “huge” and “big” can be assigned certain values. By the placement of these words in a message, the “encoded” data can be delivered to the recipient. However, this method may make it difficult to put together a meaningful, unsuspicious message.

The hidden data within these text files are very useful when wanting to transmit a message to another source. However, these techniques can also be used to fingerprint documents. As documents can be easily passed around, either through the office or on the Internet, these data hiding techniques can protect documents from being illegally copied and distributed.

For example, if Person A wrote a document that s/he only wanted Person B and C to have access to, Person A can use two different coding techniques or different versions of the same technique on the documents before handing them over to B and C. If the document ever got into circulation without A’s permission, the culprit could easily be found by comparing the copy in circulation to the ones that A handed over to B and C. Unless B or C altered the encoded data before copies were made, then the copy in circulation should match either B’s or C’s copy.

Data-Hiding in Images

Text files are not the only files that can be used for host data. Images are also another popular source for hidden data. In order to understand image steganography, an understanding of a computer image must be developed. A computer image is an array of numbers that represent light intensities at various pixels [27]. A typical image may consist of 256 colors and contain 8 bits per pixel. However, 24-bit per pixel images do exist and may also be used for data hiding.

When deciding what type of image to use for data hiding, there are many factors to consider. For example, when using an 8-bit per pixel image, a palette of 256 shades of gray is more appropriate to use because the change in colors is less distinguishable. If a 256-color palette is to be used instead, then an image without large areas of solid color is desired so the change of a pixel would not be obvious to the viewer. Due to the sensitivity of dealing with 8-bit pixel images, 24-bit pixel images are often favored in steganography since these images provide more space for hiding information [27].

Regardless of how many bits per pixel there are in an image, the same techniques can be applied to these images to hide information. These techniques include least significant bit insertion and the use of algorithms and transformations.

Least Significant Bit insertion

Least significant bit insertion, or LSB, is one of the most common techniques used to hide information in images. When working with 24-bit pixel images, three bits can be encoded into each pixel. Because the least significant bits are the ones being altered, the change is difficult to determine by the viewer. However, when working with 8-bit pixel images, this method becomes harder to implement because a change of a bit may result in a change of an entirely different color. This is why using a palette with shades of gray is recommended when dealing with 8 bit images.

Although this technique is popular due to its simplicity, it is also one of the easiest methods to accidentally alter. When transforming images to different formats, such as from GIF to JPEG, lossy compression occurs. Lossy compression, which is standard for JPEG images, uses high compression [27]. However, this high compression may cause some change from the original image. Although it is almost an

exact replica of the original image, the bits from the original image cannot be guaranteed. Because of lossy compression, GIF to JPEG transformations may cause the encoded data to be lost.

Algorithms and Transformations

Other compression algorithms and transformations are also used when dealing with images and their usage in hiding data. Some of the more popular methods are the Patchwork method, the discrete cosine transform or DCT, and the Fourier transform.

The Patchwork method takes advantage of the fact that the human eye cannot easily detect varying amounts of light [16]. The Patchwork method gets its name by "using redundant pattern encoding to repeatedly scatter hidden information throughout the cover image, like patchwork" [27]. One advantage of this technique is that it can hide a small message many times throughout an image. Because of this, even when an image is cropped or rotated, the chances of one instance of the encoded message still being intact are very high.

However, in other compression algorithms, message retention is not always guaranteed. As stated earlier, when images such as JPEG images are compressed, data may be lost. There are many different compression algorithms that perform this type of lossy compression such as the discrete cosine transform, the wavelet transform, and the Fourier transform.

The discrete cosine transform (DCT) is an algorithm that finds a set of coefficients that allow a small set of cosine functions to approximate a portion of the image [28]. For example, the JPEG algorithm uses 8x8 blocks of pixels and fits them with a set of cosine functions that can approximate a section of the image. The DCT

finds a different coefficient for each function so that the weighted sum of the functions adds up to recreate the original 8x8 block of pixels [28]. The wavelet transform and Fourier transform are both forms of the DCT. Both methods use complicated mathematical formulas in order to find the coefficients in which to map a signal into the frequency domain [16]. More information on the different mathematical formulas can be found in [16] and [28].

Data-Hiding in Audio

Audio files can also be used to hide information. With programs such as Napster, steganography is often used to copyright audio files to protect the rights of music artists. Techniques such as least significant bit insertion, phase coding, spread spectrum coding, and echo hiding can be used to protect the content of audio files. The biggest challenge all these methods face is the sensitivity of the human auditory system or HAS [16]. Because the HAS is so sensitive, people can often pick up randomly added noise making it hard to successfully hide data within audio files.

To fully understand the different techniques of hiding information in audio files, transmission of audio signals must first be understood. When working in audio environments, the digital representation of the audio and the transmission medium must always be considered.

The digital representation of an audio file is composed of the sample quantization method and the temporal sampling rate [2]. The most frequently used format for representing samples of high-quality digital audio is a 16-bit linear quantization [2]. Windows Audio-Visual, or WAV, files uses this format for their files. The temporal sampling rate determines an upper bound on the usable portion of the

frequency range. Common temporal sampling rates range from 8 kHz to 44.1 kHz. As sample rate increases, so does the amount of usable data space for information hiding.

The transmission medium of an audio signal refers to the environment in which a signal might go through to reach its destination. Bender and his colleagues categorize the possible transmission environments into the four following groups [2]:

- Digital end-to-end environment where the sound files are copied directly from one machine to another.
- Increased/decreased resampling environment where the signal is resampled to a higher or lower sampling rate.
- Analog transmission and resampling where a signal is converted to an analog state, played on a clean analog line, and resampled.
- "Over the air" environment where the signal is played into the air and resampled with a microphone.

By understanding the different mediums in which audio signals may travel, the appropriate technique for embedding data in audio files can be determined.

Least Significant Bit Insertion

Like text files, the least significant bit insertion method can also be used to store data in the least significant bit of audio files. For example, in a 16-bit per sample file, the least four bits can be used for data hiding. However, like text files, by using this method, the hidden data can be easily destroyed and detected. Resampling and channel noise may alter the hidden data, while changing the least significant bit may introduce audible noise [27]. Information may also be destroyed through compression, cropping, or A/D, D/A conversion [29]. Although this technique is simple to perform, its lack of dependability makes other methods more appealing.

Phase Coding

Phase coding is another technique used to hide data in audio files. This is done through substitution of the phase of an initial audio segment with a reference phase that represents the data. The phase of the following segments is adjusted accordingly to preserve the relative phase between segments [29]. The steps to phase coding are as follows [4]:

- The original sound sequence is broken into a series of N short segments.
- A discrete Fourier transform is applied to each segment.
- The phase difference between each adjacent segment is calculated.
- For segment S_0, the first segment, an artificial absolute phase P_0 is created.
- For all other segments, new phase frames are created.
- The new phrase and original magnitude are combined to get a new segment, S_n.
- The new segments are concatenated to create the encoded output.

In order for the receiver to decode the hidden data, one must know the length of the segments, the discrete Fourier transform points, and the intervals in which the data are hidden. This synchronization is usually determined ahead of time. Phase coding is one of the most effective schemes in terms of the signal-to-perceived noise ratio because listeners often do not hear a difference in the altered audio file when the phase shift is smooth [29].

Spread Spectrum Coding

Spread spectrum coding can also be used to hide data in audio files. Usually when audio files travel through communication channels, the channels try to

concentrate audio data through narrow regions of the frequency spectrum in order to conserve bandwith and power [2]. However, this technique requires the embedded data to be spread across the frequency spectrum as much as possible. Unlike the LSB insertion, spread spectrum coding uses the entire spectrum of the file to embed data [9].

There are many methods that can be used to spread the embedded data over the frequency spectrum. Direct Sequence Spread Spectrum encoding spreads the signal by multiplying it by a certain maximal length pseudorandom sequence called chip [2]. Unfortunately, like the LSB method, DSSS may add random noise that the listener can detect. For Frequency Hopped Spread Spectrum encoding, the original audio signal is divided into small pieces and each piece is carried by a unique frequency [29]. The main advantage of using spread spectrum coding is its resistance to modification. Because the embedded data is spread throughout the cover data, it would be difficult to modify the embedded data without causing noticeable harm to the cover data.

Echo Data Hiding

Echo data hiding hides data in a host signal by introducing an echo [16]. The embedded data is hidden by varying three parameters of the echo: initial amplitude, decay rate, and delay [16]. As the timing between the original signal and echo decreases, the two signals may blend, making it hard for the human ear to distinguish between the two signals. The value of the hidden data corresponds to the time delay of the echo and its amplitude.

By using different time delays between the original signal and the echo to represent binary one or zero, data can be embedded into the audio file. To embed more than one bit, the original signal is divided into smaller segments and each segment can

then be echoed to embed the desired bit. The final cover data consists of the recombination of all the independently encoded segments [16]. Echo hiding works particularly well with high quality audio files. Audio files with no additional degradation and no gaps of silence are preferred when using this technique [27].

Comparison of Techniques

With the many different methods of embedding data into a file, each case must be examined individually when trying to find the best watermarking technique for that file. As with every project, the trade-offs must be considered. Before implementation, the key factor must be predetermined, whether it is efficiency, speed, or robustness.

Watermarks can be further classified by the way they are embedded. Spatial watermarks are embedded directly into the pixels of the image [29]. Spatial watermarks include watermarks created using the LSB insertion or line-shift coding techniques. By using spatial watermarking, a large amount of data could be encoded within a file. However, these types of watermarks are relatively weak and not very robust. They can easily be destroyed through the slightest modification.

Spectral watermarks use the transform coefficients of the image to embed the watermark [29]. Any transform algorithm, such as the discrete cosine, Fourier, or wavelet algorithm, can produce these coefficients. Spectral watermarks are usually much more robust than spatial watermarks. Because the use of coefficients allows each segment to represent an approximation of the original data, spectral watermarks are much more resistant to tampering. Even when the cover data is rotated or cropped, the numerous approximations can still produce the original watermark when decoded. However, despite its robustness, the quality of the cover data may not be as good as the

original. In some cases, the approximations of the coefficients are not close enough and degrade the quality of the cover data [29].

Watermark Attacks

Despite the attempts to correctly identify which watermarking technique is most appropriate to use for a certain file, a watermark may not always be protected against malicious attacks. Watermarks may be detected, destroyed, or modified by an attacker. Various attacks can also be used to test the robustness of watermarking techniques. A watermark is considered robust if the watermark can still be successfully detected even after the attack severely degrades the cover data.

These types of attacks can be organized into four general categories, organized by the way in which the attacks try to defeat the watermarking. Although watermark attacks can often be classified under more than one category, they can usually be broken down into the following groups: simple attacks, detection-disabling attacks, ambiguity attacks, and removal attacks [11].

Simple Attacks

One form of watermark attacks is the simple attack. The goal of a simple attack is to impair the embedded data by manipulating cover data without an attempt to identify or remove the watermark [11]. Depending on what type of host data is being used, other possible names for this type of attack may include waveform attack or noise attack. Examples of this type of an attack include JPEG compression, addition of noise, addition of an offset, linear and general non-linear filtering, wave-form based compression, and quantization in the pixel domain [11].

Detection-Disabling Attacks

Detection-disabling attacks, also known as synchronization attacks, are another form of watermark attacks [11]. They try to break the correlation between the embedded data and the cover data to make watermark recovery difficult. Through geometric distortion, this type of attack also attempts to make the recovery of the watermark impossible for a watermark detector. These geometric distortions may include zooming, a shift in spatial or temporal direction, rotation, cropping, pixel permutations, sub-sampling, removal or insertion of pixels, or any other geometric transformation of the data [11].

Mosaic Attack

The mosaic attack is one form of a detection-disabling attack [11]. A mosaic attack on an image may consist of chopping the image into distinct sub-images. The original image can be restored by viewing the sub-images in the proper order or in a side-by-side configuration. Viewing the sub-images in this format can give the same effect as looking at the original image. Because the chopping of the image distributes the data's watermark into many pieces, the watermark becomes hard to recover. Often times, these attacks are less successful because a typical property of this attack is that the watermark remains in the altered cover data. With help from some additional computing and increased intelligence, the embedded data can still be successfully recovered.

Jitter Attack

Another popular form of the detection-disabling attack dealing with audio content is the jitter attack [1]. A jitter attack is a simple scheme that introduces noise into the protected file. In one example of a jitter attack, the audio signal was split into segments of 500 samples. A random sample from each segment was then either duplicated or deleted from each segment, resulting in segments of either 499 or 501 samples. When the segments were rejoined, the jitter produced by the attack prevented the mark bits from being located, rendering the watermark useless [1].

Ambiguity Attacks

Confusion attacks, deadlock attacks, fake-watermarking attacks, and fake-original attacks can all be referred to as ambiguity attacks. Ambiguity attacks are attacks that attempt to confuse by producing fake original data or fake cover data [11]. An example of this type of attack is the IBM attack which attempts to discredit the authority of the watermark by embedding one or several additional watermarks. The embedding of these multiple watermarks make it unclear to which was the first, authentic watermark of the data owner [11]. Another example, the fake-watermarking attack, can be implemented by removing the original watermark and replacing it with a new bogus watermark to erase evidence of previous ownership.

Removal Attacks

Removal attacks attempt to analyze the cover data, estimate the watermark or the host data, separate the cover data into host data and watermark, and discard only the watermark, leaving the original data vulnerable to illegal use [11]. Examples of this

type of attack are collusion attacks, denoising, inversion attacks, certain non-linear filter operations, or compression attacks that use synthetic modeling of the image [11]. For example, in an inversion attack, a malicious attacker may gain knowledge of the technique used to embed the data. The attacker can simply reverse the insertion process to perfectly remove the watermark and use the newly unprotected content. In a type of collusion attack, an attacker may use several copies of the same data, each with a different watermark, to construct a single copy containing no watermark. Also under the classification of removal attacks are attacks that are tailored to a specific watermarking scheme and combat it by exploiting conceptual cryptographic weaknesses of the scheme that make it vulnerable to a specific attack [11].

Legal Attacks

One type of attack that resides outside the realm of those four categories is legal attacks. Legal attacks do not require any technical support or methods. Instead, legal attacks use other sources of information such as legal precedents or the identity or reputation of the content owner to establish doubt of ownership in court. Legal attackers use the justice system to determine whether a watermark actually constitutes the proof that the owner claims [12].

The transition of attacks between the classifications may be unclear at times. As stated earlier, the properties of some attacks may allow them to be classified under more than one category. For example, cropping can be considered either a simple or detection-disabling attack. Denoising and some non-linear filtering attacks can be considered either simple or removal attacks [11]. With all the overlapping similarities

and gray areas between these groups, individual discretion should be used when classifying different attacks.

Steganography Software

With the knowledge of different watermarking techniques and attacks that watermarks may encounter, software can be produced to implement and test these watermarks. Software can also be designed for purposes of steganalysis, the science of detecting hidden communications [23]. Neil Johnson, a researcher at George Mason University in Virginia and associate director of GMU's Center for Secure Information Systems, is one of the growing numbers of computer scientists working in the field of computer steganalysis. "I analyze stego tools. I try to find out what can be detected or disabled. I see what their limitations are," states Johnson [23].

Software for Testing

Some of the tools that Johnson refers to are software applications that test the robustness of watermarks. StirMark and unZign are examples of this type of software. They are examples of software that attempt and are often successful in removing copyright information and other types of watermarks from files. Like many other programs that break established security mechanisms, these programs are intended to demonstrate the weaknesses in current algorithms so that companies will be motivated to develop more robust watermarking technologies for their products [22].

unZign

unZign is used on watermarked JPEG pictures and can be downloaded off the Internet for public use. unZign uses pixel jittering along with a slight image translation

to attack the image being tested [26]. Depending on the watermarking technique used on the data, unZign is most often successful in efficiently removing or destroying the embedded watermark. The most recent version of unZign and unZign 1.2 can be used with Linux and Windows platforms.

StirMark

StirMark is another software tool that tests the robustness of image watermarking techniques. StirMark was created in order for companies to have their watermarking product tested by a trusted third party. With the release of StirMark 3.1, the first benchmark for watermarking was made possible [25]. The benchmark is then used to subject the watermarked image to several attacks to which the watermark should be able to defeat. Unlike unZign, which uses the jitter attack, StirMark applies minor geometric distortions to test the watermarked images. When applying StirMark to an image once, no noticeable loss of quality is perceived. However, because the StirMark benchmark service automatically starts hundreds of tests on the images, existing failures are sure to be brought to attention [25].

The implementation of StirMark testing is done through three steps and is very simple to follow. First the client sends a library of images to be evaluated along with the specification of the evaluation profile and level of assurance to be used. Then the program begins performing the hundreds of tests on the library using its library of images. When the tests are done, the results are sent back to the client. The simplicity of this process is one of the many features of StirMark. Other advantages of using StirMark include [25]:

- Simple interface with watermarking libraries

- Ability to use different evaluation profiles and strengths
- Ability for client to submit libraries for different platforms (Linux and Windows)
- Evaluation procedures, profiles, and code are all publicly available
- Ability to encrypt the results of the testing

Often times, the results of the tests performed by either unZign or StirMark prove that there is still much to be developed in the world of watermarking. However, this is not a surprise to Johnson and other researchers. According to them, "Current stego programs don't work well at all. Nearly all leave behind fingerprints that tip off a careful observer that something unusual is going on" [23].

Software for Implementation

Although there is no steganography software that is completely safe from watermark attacks, there are many programs currently on the market or available on the Internet that provide content owners some sense of security. These programs may differ by the content that they work with, the platform on which they can be used, and other criteria. Some of the more popular programs include MandelSteg, Steg, S-Tools, Stego, and Hide and Seek.

MandelSteg

Mandelbrot steganography, or MandelSteg for short, was developed by Henry Hastur [4]. This program produces a .GIF file as the resulting image by using a generated Mandelbrot fractal with the hidden message within the fractal [1]. Like the watermark testing software, MandelSteg is easy to use. The package consists of two executables, one executable to create the fractal and embed the information and the

other to extract the hidden from the fractal [4]. With the use of certain command flags, data can be stored simply in the specified bit of each pixel. Often times, MandelSteg can be regarded as a one-time pad with the Mandelbrot image as the pad and the coordinates and size of area generated as the key [4].

Steg

Steg, designed by the JPEG group, compresses image files using the JFIF format of the JPEG standard and embeds the data to be hidden into the JPEG file during the compression process [28]. This watermarking can be retrieved during the decompression process. Like MandelSteg, Steg also consists of two executables that hide and recover embedded information. One executable compresses image files using the JPEG standard and does the data embedding while the other decompresses a JPEG file and retrieves the embedded data. Steg implements the encoding procedure by using the discrete cosine transform on 8x8 blocks of pixels [28].

S-Tools

Designed by Andrew Brown, S-Tools, short for Steganography Tools, is a program that allows users to hide information within different types of files including, Windows sound files, otherwise known as .WAV files [28]. With a user-friendly interface, messages can be hidden and retrieved from .WAV files with a click of a mouse. By using least significant bit insertion, S-Tools is able to hide information within audio samples without changing the size of the sound file. This technique is also chosen because the information hidden inside the .WAV files will not sound any different to the human ear than the original .WAV [4].

One feature that S-Tools contains is the use of a random number generator that is based on the output of the MD5 message digest algorithm [4]. S-Tools uses this number generator to pick the order of the bytes used in the embedding process. This is done to defeat attackers who might be applying statistical randomness tests on the lower bits of the sample to determine whether there is information hidden there. With this feature, S-Tools ensures a little bit more security to its product.

Stego

For Macintosh users, Romana Machado developed Stego, a tool that hides data in the Macintosh PICT format files, without changing the appearance or size of the PICT file [4]. Like S-Tools, Stego has a user-friendly interface that can easily hide or retrieve information from an opened PICT file. Stego also uses the least significant bit insertion method to hide data within each of the RGB color values. However, unlike S-Tool's use of a random number generator, Stego inserts the data to be hidden in sequential order into the PICT file. Stego can hide data in 8, 16, or 32-bit Macintosh PICT files [4].

Hide and Seek

Developed by Colin Maroney, Hide and Seek embeds and retrieves information from GIF files. Available for both Windows and DOS systems, the user-friendly interface of this software application makes it easy for users to perform watermarking. One feature of Hide and Seek is the ability to add encrypted header information within the image at the time of watermarking [4]. The encryption of the header information is performed by the IDEA cipher and is an optional feature provided by the program. The

information stored within the header is a seed for a rand() function, the length of the secret information, and the version number of the program [4]. The header may also contain any additional information the artist may want the user to know about the data. Like other tools, Hide and Seek uses least significant bit insertion along with random dispersion to hide data within images.

Other Software

There are several other steganography tools available to those who want to protect their content before making it available on the Internet. For some programs, a full-featured trial version is available for a limited time to potential buyers. All of these software tools can be purchased through the companies who develop them, commercial vendors, or via the Internet. Most software packages are easy to use and explained thoroughly by either providing paper documentation or a README file. Although only some of the tools are mentioned above in greater detail, there are other programs on the market that deserve mention.

Other watermarking tools available for the Windows environment include [19]:

- **Steganos 3 Security Suite:** hides data in graphic and sound files and is available in English, German, and Italian
- **Digital Picture Envelop:** a BMP-based stego program that can hide a very large amount of data in a single file
- **MP3Stego:** hides information in MP3 files and produces a near CD quality sound
- **S-Mail:** uses strong encryption and compression to hide data in .EXE and .DLL files, both Windows and DOS versions are available
- **SubiText:** subtly changes text features, such as narration tense, perspective, and single word synonym replacements to hide data within text files

Steganography tools are also available for the Unix environment. Some of these include [21]:

- **StegParty:** a text-based tool that utilizes small changes in spelling and punctuation
- **Visual Cryptography:** hides data in two transparent images, the hidden data is revealed when the two images are stacked
- **Snow:** conceals messages in text files by appending tabs and spaces on the end of lines
- **Nicetext:** pseudo-random text-based steganography tool using context-free grammar and customizable dictionaries

There are also many watermarking programs written for Macintosh users. Some of more popular programs are [20]:

- **Paranoid:** primarily an encryption program that encrypts files using IDEA, or triple DES, also a steganography program that hides files in sounds
- **Mimic Functions:** pseudo-random text-based watermarking program that uses context free grammar to hide data, by using the customizable dictionaries, files can be hidden within text that sounds like Shakespeare, Aesop's Fables, and other interesting styles
- **FatMacPGP 2.6.3:** is the most recent version of MacPGP optimized for PowerMacs, contains Stealth option leaving encrypted data in format for steganographic use

Software Companies

There are several companies that produce the many different software packages available for digital watermarking. Similar to any type of product development, some steganography tool developing companies are more prominent than others.

Digimarc, one of today's leading manufacturers of digital watermarking technologies, built its first product in 1996 and has been one of the most important steganography software suppliers ever since. Their product allows digital data to be

invisibly embedded into traditional and digital content, including movies, photos, graphic images, and even documents such as passports and event tickets [5].

Digimarc is also the leading provider of solutions to counterfeiting and piracy over today's Internet. They provide anti-counterfeiting solutions to the world's leading central banks and image commerce technologies to leading stock photo agencies and major corporations. "Our digital watermark technology not only protects online copyrights, but also connects content creators and buyers to facilitate commerce and promote collections. As analog and digital media continues to converge, we want our services to be more accessible to benefit graphic artists, commercial photographers and the firms that license their work," states Digimarc CEO, Bruce Davis [6].

Other software companies include Blue Spike, Inc., Cognicity, Signum Technologies, MediaSec Technologies, Alpha Tec. Ltd., and many others. Like Digimarc, these companies provide users with tools that implement watermarking schemes on different types of multimedia content. Information on their specific products and availability can all be obtained from their respective webpages.

Watermarking Applications

With all the steganography software available for public and commercial use, many companies and other interest groups are beginning to use watermarking tools to protect their content. For example, the Vatican, Playboy, Corbis, and Pitney Bowes are all beginning to use steganography software to protect their content. Many product vendors, such as Kodak, are also beginning to implement this software within their own products.

Vatican Library Project

Nearly one and a half million books are stored at the Vatican Library. Virtually all civilizations and cultures in the history of humanity are represented somewhere in the Vatican Library. However, due to distance and cost, only few scholars around the world are able to make use of this vast knowledge. In order to make this information more accessible to those world wide, IBM, the Vatican Library, and the Pontifical Catholic University have joined together to form the Vatican Library project. Using the latest technology, the project is making available, via the Internet, digital replicas of selected manuscripts from the Vatican Library [13].

The Vatican has begun using visible watermarks as copyright notices on the manuscripts they post on the Internet. This technique was developed at the request of the Vatican Library as part of a project that protected the manuscripts they made available through the Internet. "The intent was to make clear, to all who would see the images, that they were the property of the Vatican Library, without detracting from their utility for scholarship" [18]. This use of the watermark, like a copyright notice, identifies the ownership of materials and reminds viewers of their limited copying rights.

Playboy

After discovering that another website had stolen some of their images and displayed them illegally, Playboy decided to use Digimarc's PictureMarc program to protect their Internet content. "We welcome new technology like Digimarc that helps us protect one of our most valuable assets, our copyrighted images," states Eileen Kent, Vice President of New Media for Playboy Enterprises [8]. Playboy now embeds every

photograph that they put on the Internet with a distinct watermarking. They also use Digimarc's MarcSpider program to search the Internet for illegal copies of Playboy's watermarked photographs [8].

Other Digimarc Ventures

Not only has Digimarc joined forces with Playboy, but they have created alliances with other companies, such as Corbis and Pitney Bowes, as well. Corbis, the provider of the Internet's largest collection of high-quality images, uses Digimarc to protect its content. "Corbis has more than 2.1 million images online, and we use Digimarc's digital watermarking software and services to manage these assets and market our collection to new media professionals. Digital watermarks are a critical component of our strategy to protect our image assets online, and to generate additional licensing agreements for our collection," states William Radcliffe, Director of Technology Development for Corbis [7].

Postage giant Pitney Bowes, together with Digimarc, is creating ways to embed digital watermarks on envelopes to bridge regular mail with the Internet [24]. Pitney Bowes launched its on-line postage service in 1999 and is looking for ways to expand their services. Having already been confronted with counterfeiting problems with their postage metering, Pitney Bowes hopes to find their solutions within Digimarc. "We believe that Digimarc digital watermark technology will offer new and exciting innovations in a range of postal and other applications," states Jim Euchner, Pitney Bowes' vice-president of advanced technology [24]. The two companies hope to work together to create digital watermarks for envelopes and add secure features for metered postage, especially for its Internet postage service.

Trustworthy Digital Camera

With the millions of images on the Internet today, it is hard to prove ownership of these images. It is also hard to tell if images are authentic and not altered or transformed in any way. Gary L. Friedman, from the California Institute of Technology, first proposed the idea of trustworthy digital camera for professional photographers [10]. This type of digital camera would embed a digital watermark into every photograph that it takes. In doing this, the camera would provide a mark of image ownership from the moment that the image was captured.

This camera would not only be able to determine proof of ownership but can effectively guarantee authenticity as well. With the advancement of today's technology, it is not uncommon for insurance claims phonography to be done digitally. The trustworthy camera can be a valuable tool for evidence collectors and its pictures used as evidence in courtroom cases [10]. The authenticity of the evidence can be guaranteed by the watermarks embedded by the trusted camera. One example of this type of camera is the Kodak Digital Science DC260 Zoom Camera which among its features, contains scripting and graphical watermarking [17].

Future of Watermarking

From head tattooing, to invisible ink, to digital watermarking, steganography has evolved with time and technology. As the needs of the users changed, so did the methods of steganography. Now with the technology of the Internet, more sophisticated implementations of steganography are necessary. The process of digital watermarking is needed to protect the huge amounts of content over the Internet. One of the most pressing issues is copyright protection. "With the ease in which individuals can

download images from the Web, digital watermarking is fast becoming a fundamental tool to enable image commerce on the Internet, providing the business with a means to post images yet still track and deter copyright infringement" states Hugh Mackworth, president of Digimarc[8].

Unfortunately, as watermarking attacks have shown, none of the techniques that are employed today are indestructible. Researchers and steganography engineers still need to develop digital watermark techniques that are as reliable as their paper ancestors were and cost-effective enough to be widely implemented on a commercial basis. Only when these watermarks are robust enough to resist all attacks will content providers be willing to commit to delivering real, high quality multimedia over public networks, such as the Internet.

However, there is no doubt that this advancement can be accomplished. With companies such as Digimarc currently developing additional applications to address other forms of visual media such as DVD, video, and other distribution channels, an improved generation of watermarking techniques is sure to arise [5]. With the way steganography has evolved from ancient times to present day, there is no telling where the future of steganography, specifically digital watermarking, will take us.

Works Cited

[1] Anderson, Ross J., Kuhn, Markus G., Petitcolas, Fabien A.P., University of Cambridge, *"Attack on Copyright Marking Systems"*, http://www.cl.cam.ac.uk/~fapp2/papers/ih98-attacks/.

[2] Bender, W., Gruhl, D., Lu, A., Morimoto, N., IBM Systems Journal, *"Techniques for Data Hiding"*, http://www.research.ibm.com/journal/sj/mit/sectiona/bender.pdf.

[3] Cox, Ingemar, J., Kalker, Ton, Linnartz, Jean-Paul M.G., and Miller, Matt L., *"Chapter 18: A Review of Watermarking Principles and Practices"*, http://citeseer.nj.nec.com/cache/papers2/cs/13887/ftp:zSzzSzftp.nj.nec.comzSzpubzSzingemarzSzpaperszSzbookchapter99.pdf/a-review-of-watermarking.pdf.

[4] Davern, P. and Scott M., School of Computer Applications, Dublin City University, "*Steganography: its history and its application to computer based data files",* http://www.jjtc.com/Steganography/bib/3000027.htm.

[5] Digimarc, *"About Digimarc",* http://www.digimarc.com/about/index.htm.

[6] *"Digimarc Announces Flexible Pricing for Digital Watermarking Software and Services"*, http://www.digimarc.com/news/pr00-13.htm.

[7] Digimarc, "*Digimarc Announces Flexible Pricing for Digital Watermarking Software and Services"*, http://www.digimarc.com/news/pr00-13.htm.

[8] *"Digimarc Technology to Help Playboy Crack Down on Image Piracy over the Internet",* http://www.digimarc.com/news/pr97-12.htm.

[9] Ferrill, Elizabeth and Moyer, Matthew, "*A Survey of Digital Watermarking"*, http://www.cc.gatech.edu/~mjm/dw/watermarking.html.

[10] Friedman, Gary L., Advanced Engineering and Prototype Group, California Institute of Technology, *"The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image"*, http://techreports.jpl.nasa.gov/1993/93-1589.pdf.

[11] Girod, Bernd, Hartung, Frank, and Su, Jonathan K., University of Erlangen-Nuremberg, *"Spread Spectrum Watermarking: Malicious Attacks and Counterattacks"*, http://citeseer.nj.nec.com/cache/papers2/cs/1836/http:zSzzSzwww.nt.e-technik.unierlangen.dezSzLNT_IzSzpublicationszSzpub_listzSzpub_fileszSzInt1999_008.pdf/hartung99spread.pdf.

[12] Holliman, Matthew, Yeo, Boon-Lock, and Yeung, Minerva M., Intel Corporation, *"Digital Watermarks: Shedding Light on the Invisible"*, http://www.computer.org/micro/mi1998/pdf/m6032.pdf.

[13] IBM DB2 Digital Library, "*The Vatican Library"*, http://www-4.ibm.com/software/is/dig-lib/vatican/vatican.html.

[14] Johnson, Neil F., Center for Secure Information System, George Mason University, "*Steganography*", http://www.jjtc.com/stegdoc/index2.html.

[15] Kahn, David. The Code Breakers. Scribner Publishing, New York, New York, May 1996.

[16] Katzenbeisser, Stefan and Petitcolas, Fabien A.P., Information Hiding: Techniques for Steganography and Digital Watermarking, Artech House, Inc., Norwood, MA, 2000.

[17] *"Kodak Digital Science"*, http://www.stealth.viaduk.net/descr/6/dc260.html.

[18] Lotspiech, Jeffrey, Mintzer, Fred, and Morimoto, Norishige, IBM Research Division, *"Safeguarding Digital Library Contents and Users"*, http://www.dlib.org/dlib/december97/ibm/12lotspiech.html.

[19] Milbrandt, Eric, *"Steganography Software"*, http://members.tripod.com/steganography/stego/software.html.

[20] Milbrandt, Eric, *"Steganography Software"*, http://members.tripod.com/steganography/stego/softwaremac.html.

[21] Milbrandt, Eric, *"Steganography Software"*, http://members.tripod.com/steganography/stego/softwareunix.html.

[22] Milbrandt, Eric, "*Watermarking"*, http://members.tripod.com/steganography/stego/watermrk.html.

[23] McCullagh, Declan,*"Secret Messages Come in .Wavs"*, http://www.wired.com/news/politics/0,1283,41861,00.html.

[24] Olsen, Stepfanie, *"Pitney Bowes Seals Digital Watermark Deal"*, http://news.cnet.com/news/0-1007-200-3684787.html?tag=prntfr.

[25] Petitcolas, Fabien A.P., Computer Laboratory, University of Cambridge, *"Evaluation of Watermarking Schemes"*, http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/benchmark/.

[26] Petitcolas, Fabien A.P., Computer Laboratory, University of Cambridge, *"Weaknesses of Existing Watermarking Schemes"*, http://www.cl.cam.ac.uk/users/fapp2/steganography/image_watermarking/unzign/.

[27] Sellars, Duncan, *"Introduction to Steganography"*, http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.html.

[28] Wayner, Peter.Digital Copyright Protection. AP Professional, Chestnut Hill, MA, 1997.

[29] Yang, Yang, Faculty of Computer Science, Dalhousie University, *"Digital Watermarking Technology"*, http://www.cs.dal.ca/~yyang/6505/6605.pdf.