I.J. Information Technology and Computer Science, **2019, 3, 23-37**
Published Online March 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2019.03.03

Attacking Image Watermarking and Steganography - A Survey

Osama Hosam
The Collage of Computer Science and Engineering in Yanbu, Taibah University Saudi Arabia.
As with SRTA-City, Alexandria, Egypt
E-mail: mohandesosama@yahoo.com

Received: 16 September 2018; Accepted: 16 January 2019; Published: 08 March 2019

***Abstract*—**Image hiding techniques include steganography and watermarking. Steganography procedures are directed to keep the secure information from eavesdropping and perturbations. On the other hand, watermarking algorithms are used for keeping the watermark robust to attacks. When the attacker tries to perturb the carrier image to remove the watermark, the image quality will be degraded to level that makes it useless. Data hiding is essential in many applications such as communication channel security, data security and forgery detection. Watermarking is used in copyright protection. Image hiding attacks can be active or passive. In active attack, the attacker changes the content of the data. While in passive attacks the attacker tries to guess the secure data by eavesdropping. In this paper, we discuss the image data hiding attacks that directed to both secure message and carrier image. First, message attacks such as Oracle and Template attacks will be discussed. Second, the carrier image attacks are presented in two broad categories, namely passive and active attacks. Finally, the paper conclusion will be presented. The paper presented image data hiding attack types in professional and well-organized categories.

***Index Terms*—**Data Hiding, Attacks, Steganography, Watermarking, Steganalysis, Passive attacks, Active Attacks, Geometric, Image enhancement, Image Degradation.

I. Introduction

Data hiding is a broad field mainly define any system with data embedded into other data. Generally, the embedding can be visual such as a movie with logo watermark, or imperceptible such as covert communication. Data hiding is defined as embedding imperceptible data into another carrier digital signal. Data hiding can be divided into two main fields, steganography and watermarking. These fields are closely related but they have few differences that affect the embedding algorithms and related attacks [1].

In watermarking systems, the secure data is related directly to the image. The company can protect the copyright of its documents by embedding their own logo into their own images [2]. In steganography systems, the secure data is not related to the carrier image. The cover image is used as a channel for covert communication [3]. For example, terrorists may use internet images as a communication media. They transfer maps and terror plans by embedding them securely into Internet images [4]. Therefore, in watermarking systems the capacity of embedding is not an issue, but in steganography the channel capacity is important. Robustness to attacks is a major research topic in watermarking. In other words, If the attacker discovers the watermark, the cover work is still useful as long as the watermark is robust to attacks. In steganography, the main objective of the attacker is to discover the secure data. For example, secure communication with steganography can be attacked with eavesdropping. Table 1 shows a short comparison between watermarking and steganography.

Table 1. The comparison between the most common data hiding techniques, namely watermarking and steganography.

	Watermarking	**Steganography**
Robustness	Active attacks	Passive and Active attacks
Embedding capacity	Low	High
Image/Message Relationship	Exist	Not exist
Imperceptibility	Not important	Very important
Message Encryption	Not Important	Very important

Watermarking procedures should be more robust to active attacks than passive attacks. Whereas steganography must be robust to both passive and active attacks. The embedding capacity for watermarking can be merely a name of the image owner (a couple of words), while steganography capacity must be higher such as hiding an entire document with thousands of words. The relationship between the embedded message and its carrier is important in watermarking but not important in steganography. Finally, the imperceptibility is not important in watermarking but very important in steganography. Python based implementation of Least Significant Bit (LSB) steganography for colored images is provided in [59].

To address data hiding attacks, In [24] year 1998, the authors introduced the attacks on watermarking systems that mainly developed for copyright protection. However, the authors didn’t cover all types of data hiding techniques such as steganography-based attacks. In [5] year 2001, the authors proposed a benchmark for evaluating different algorithms of image watermarking. The author proposed Stirmark attack which is designed to resemble printing and scanning watermarked documents. The robustness of this attack means robustness to vast category of watermarking attacks. Stirmark will be explained in detail in section 2. As the authors stated, Stirmark is limited in its ability to impair sophisticated watermarking attacks.

Johnson and Jajodia [50] have explored steganography and steganalysis techniques. At the end of their book, they proposed selective countermeasures to defend against steganalysis. For example, stego-images can be protected by using their own fingerprint.

A general framework for robustness of digital watermarking against attacks is proposed in [55] year 2003. The authors payed attention to the attackers according their knowledge of the watermarking scheme. They divided the attacks to two types, fair and un-fair attacks. In the fair attack, the attackers use the publicly available information about the watermarking scheme to commit his attack. In un-fair attack, the attacker collects information about the watermarking algorithm using illegal methods. In addition, the authors introduced mathematical framework inspired by information theoretic principals stating that the security of any watermarking procedure in case of un-fair attack can be quantified.

In reference [52] year 2004, the authors divide the information hiding techniques into two broad categories; spatial domain and transform domain. The authors successfully proposed counter measures for each type of embedding process found in literature. The authors also mentioned references to the latest hiding techniques such as EzStego, F5, Hide and Seek, Hide4PGP, Jpeg-Jsteg, Mandelsteg, OutGuess, Steganos, S-Tools, and White Noise Storm. In addition, steganalysis tools such as RS-steganalysis [21], PoV-based, Chi-square, Palette checking, RQP method, and Histogram analysis are presented with the corresponding targeted data hiding technique. However, their approach lacks the definition of geometrical attacks and their destructive effect – more detail on PoV-based, Chi-square and Histogram modification attacks will be provided in this paper.

Licks et al. (2005) [26] presented different types of geometric attacks. The authors stated that geometric attacks can disturb vast category of watermarking systems and change them into useless algorithms. The attacks can be intentional or unintentional, such as scaling, rotating, cropping and changing aspect ratio. Data hiding algorithms can be designed to circumvent specific type of geometric attack but not all attacks. The authors mentioned only geometric attacks and didn't cover vast majority of attacks.

A comprehensive survey on watermarking security is introduced in [53], year 2006. The authors covered the theoretical side of watermarking security. They introduced formal measures to introduce standard and formal model for assessing security of the watermarking algorithm. The practical side is also introduced. Test is performed on vast majority of watermarking techniques to evaluate their performance and introduce suitable and effective countermeasure. However, the authors stated that, they didn't encompass steganography techniques.

Cox. et al. (2007) [1] divides the attacks according to the knowledge available to the attacker. The authors divided the adversaries into four categories:

- The attacker doesn't know any knowledge regarding the algorithm and doesn't have a watermark detector algorithm. In this case the attacker may try different geometric and degradation attacks. For example, the attacker can compress, resize, or filter the attacked image.
- The attacker may have many watermarked images containing the same watermark. An attack such as "Collusion" attack might be suitable for completely removing the watermark without even knowing about the watermarking algorithm.
- The attacker is assumed to have knowledge about the algorithm but doesn't know about some keys. The attacker may search for vulnerabilities in the detection process using the masking attack.
- The attacker may obtain a detector but doesn't have the algorithm. In this case he/she would try different perturbed work until he/she gains good assumptions about the detection algorithm. The attacker can then use this knowledge to commit attacks such as Oracle attack.

Good classification of image watermarking attacks is provided in [58, 5]. Watermarking attacks are divided into removal, cryptographic, geometry and protocol attacks. The authors didn't cover passive attacks which is mainly found in steganalysis approaches. However, in our proposed classification, passive attacks are also covered.

Tanha et al. (2012) [51] categorized attacks into two categories. The first category is the unauthorized action-specific attacks. For example, copy attack, collusion attack and oracle attack – more detail on these attacks is provided in the next section. The second category is system attacks, an example of this category is mosaic attack.

The authors in [56] year 2016, proposed methods for measuring the security of watermarking schemes, they provided guidelines of how to propose robust and non-breakable watermarking scheme. In addition, they provided methodologies of attacking popular watermarking schemes. They focused on both sides, the attacker side and the defender side.

The authors in [57], 2018, unified the attacks on both machine deep learning and watermarking. They proposed a black-box attack model that works well for both adversarial machine learning and watermarking attacks. The authors main object is to collect efforts in both fields and come up with more robust and non-breakable shield.

In this paper, a comprehensive survey on image data hiding attacks will be introduced. As shown in Figure 1, watermarking and steganography have two parts of data.

The first part is the carrier image. The second part is the security message. First, we introduce attacks directed to the security message. The security Message is called watermark or logo in some data hiding techniques. We start with message removal attacks namely collusion, denoising, remodulation and quantization attacks [5].

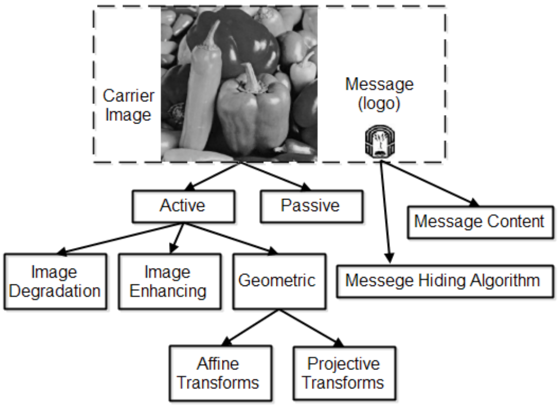


Fig.1. the organization of attack types. There are attack types related to the carrier image and other attacks directed to the embedded message.

Desynchronization attack is a technique used to disturb the watermark extraction process. The main objective is to remove the synchronization data assigned by the embedding algorithm, the synchronization data is the main reference for the extraction process [8]. In Oracle attack, the attacker tries to build an extraction algorithm [1]. With protocol attack, due to the versatility of image data, the attacker can extract his own watermark from the image and allege copyright [8]. With collage attack, the attacker uses already authenticated image and perturb it with adding parts of another authenticated image for the same author [9]. The inverse transform is applied to the image to extract the watermark with template attack [10].

The carrier image-based attacks are classified into passive and active attacks. Passive attacks try to detect the existence of an embedded data, and/or extract it without perturbing the carrier image [11].

In active attacks, the attack objective can be malicious or non-malicious [12, 36]. In malicious attacks the attacker finds methods to remove the watermark with keeping an acceptable level of the carrier image quality. In non-malicious attacks, the image maybe compressed, rotated translated or resized for application purposes. Even though non-malicious operations are simple and applied on mostly every image, these operations are reported to be destructive to most data hiding techniques [4].

Active attacks are broadly divided into image degradation image enhancement, and geometric attacks. In image degradation attacks, the image is exposed to noise addition and perturbation. In malicious type of attacks, the attacker tries to perturb the image to an acceptable degree to remove the embedded data. In image enhancement, the attacker enhances the image by removing image noise. The embedded data is considered an image noise and is removed with noise filters [13].

In geometric attacks, the attacker changes the overall structure of the image. The attacker changes – for example, the image size by resizing, or the image shape by shearing, or the image content by cropping, and so on [14, 15]. Geometric transforms can be affine. Affine transforms keep parallel lines parallel and keep the angles. While in projective transforms, the transform keeps lines but not essentially keeps parallelism and angle values. Examples of affine transforms are rotation, scaling and translation. Examples of projective transforms are shearing, warping and perspective transform.

The paper is organized as follows; in section 1 we introduce message-based attacks, in section 2 we present carrier signal attacks. Finally, we propose the paper conclusion and future trends.

II. Message Attacks

The secure message has versatile formats, the message maybe in a form of text, image, watermark, or any digital data. The extraction of the secure message or distorting it is an objective for many data hiding attackers. The attacker may intercept the covert channel and remove the secure message. The extraction process must be maintained with perfect message/carrier synchronization. If there is a method to tamper that synchronization, the attacker will be able to stop the secure communication. Another type of attacks is intended to gain the security key in order to discover the secure information or commit an active attack. With weak hiding procedure, watermarking data can be copied form one image to another [5].

A. Message Content Attacks

The message content attacks aim to remove or perturb the embedded message. The message maybe removed, manipulated or guessed. Noise is added to the image for removing the watermark. In addition, the encryption key can be guessed by the attacker to decrypt the secure message. In this section, the attacks directed to removing the embedded message are explored.

A.1. Message Removal

The message is removed partially or completely from the carrier without the need of the security key. After the attack, any hiding algorithm will not be able to extract the watermark. There are many categories of attacks for removing the message. They can be broadly classified to denoising, quantization, collusion and remodulation attack. In denoising attack, the objective is to keep the quality of the message carrier while trying to remove the message. In the denoising processes, the carrier image is considered a signal and the watermark, or the message is considered a noise. The objective is to remove or reduce the noise.

In quantization attack, for example the quantization step in JPEG compression, the attacker objective is to restore the original quantization table of JPEG compressed carrier image. JPEG compression starts with converting the color system of the image from any color system such as RGB to YUV [6].

The image is then divided into 8x8 blocks, DCT coefficients are calculated for each block. Then the quantization table is used to quantize the DCT coefficients, the middle frequency coefficients are used in hiding data (quantized) because they have less distortion effect on the image. After quantization, the image is distorted, and its quality will be acceptably degraded, therefore data hiding is done in the quantization step.

The last step is called the coding. Coding step is considered the main step in image compression. Huffman coding is adopted in JPEG image compression. Some methodologies are implemented specifically to withstand JPEG compression attack [16].

Collusion attack can't be used with single carrier image, instead many carrier images with the same message are needed to be able to partially or completely remove the embedded data. The more images obtained, the higher probability of restoring the original image and removing the watermark. Restoring the original image can be achieved also with different messages in the collected images, but the restoration process will be harder. The carrier image is restored by averaging the provided image collection or obtaining the original image from different parts of the image collection.

Results showed that in average, if the attacker obtained very small number of carrier images such as 10 images, the attacker will be able to completely remove the embedded message.

A common scheme for remodulation attack [6,17] is to use a modulation procedure in the extraction which is opposite to that used in embedding. Assume a good estimate of the embedded data can be obtained, then we can easily detect the secret data by subtracting the median-filtered estimated data from the carrier image. The extraction procedure can be made robust to correlation-based detection by subtracting resized by a factor of 2, and high-pass filtered version of the secure data.

A.2. Protocol Attack

In protocol attack, the entire embedding concept is attacked. The attack is applied specifically to watermarking to allege copyright. The carrier image is rich with versatile data, any watermark has high probability of being part of the carrier image. The attacker extracts his own watermark from the image and allege the ownership. This attack creates ambiguity with respect to the true ownership of the carrier data. Therefore, new concept for watermarking is proposed [5, 8] in which the watermark must be noninvertible to overcome the protocol attack. The watermark is noninvertible only if the watermark can't be extracted from the carrier document. To achieve noninvertible watermarking, a one-way function can be used in the embedding process.

A.3. Collage Attack

The collage attack occurs when the attackers knows the watermark. This attack is firstly proposed by Holliman et al. [9]. There are two types of Collage attack [19], the first attack copies part of the watermarked image to an arbitrary position in another authenticated image. Since both images are watermarked by the same watermark, the newly forged image will pass security check. The second type combines parts of the watermarked images of the same owner and forge new image by combining those parts and keep their relative positions in the original images. Figure 2, shows an example, there are two authenticated source images, one image is shown in Figure 2 (a) and another image contains an animal. Both images are watermarked with the same watermark. A third image is shown in Figure 2 (b) is forged from both images. The animal position is the key to define the type of collage attack. In the first type, the animal location in the newly forged image is an arbitrary position. In the second type, the animal location in the newly forged image is the same as its location in the source image.

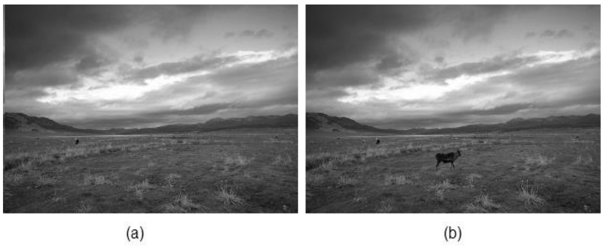


Fig.2. Collage attack (a) original watermarked image (b) tampered image from two watermarked source images. In the first type, the animal in (b) should be in an arbitrary position other than its original position. In the second type, the animal should be in the same location as the source image it is taken from.

B. Message Hiding Algorithm Attacks

The attack is directed either to the message itself (as shown in the previous section) or the embedding algorithm. The embedding algorithm is attacked by guessing or rebuilding the extraction decoder. Here the most common algorithm attacks are explained.

B.1. Message Extraction Desynchronization

In this scheme the objective is not to remove the secure data. Instead, the extraction synchronization process is tampered. For example, suppose that the embedding process of an image is done pixel by pixel and there is a reference pixel in the image. That reference pixel is used in the pixel by pixel embedding process. If reference pixel related values are changed by tampering the image, the extraction process will not succeed, i.e., the synchronization process will fail. The attacker can detect the secure data if he/she is able to get perfect synchronization in the extraction process. However, far from the na ïve synchronization example provided above, the synchronization process is complex and impractical [5].

B.2. Cryptographic or Oracle attack

Cryptographic attack is mainly used with public embedding process. The attack tries to intercept the communication to recognize the embedding key [54]. The key is then used to discover the secure message and then remove it. The attacker can also change the secure data and embed misleading data instead. One approach to commit cryptographic attack is brute force guessing of the secure data. However, the brute force approach is computationally complex. Another approach is called Oracle attack, in which the attacker is trying to build the detection algorithm or a public decoder. The secure data can be removed by tempering the carrier image keeping an acceptable quality level until the decoder can't detect the watermark. A straight forward defeat for Oracle attack is to randomize the extraction procedure [7, 18].

B.3. Template Attack

Templates are frequency domain transforms used as a reference for discovering the embedding procedure. After estimating the embedding procedure, the inverse transformation is applied to detect the embedded watermark. The template attack contains two phases; in the first phase, a denoising filter such as median filter is applied on the stego-image to estimate the original clear image. The stego-image is subtracted from the estimated image to define the difference. In the second phase, Fourier transform is calculated for the carrier image, peaks in the transform is defined. The amplitude of the identified peak modified according to the results from phase 1. Finally, the inverse Fourier transform is calculated to get the embedded watermark [10].

B.4. Sensitivity attack

The sensitivity attack assumes the knowledge of the extraction scheme (detector) and the availability of a watermarked image. The correlation between the watermarks and the carrier image is compared with specific threshold in order to discover the existence of the watermark or not. The decision boundary between the "watermarked" and "not watermarked" is hyperplane [41].

III. Carrier Image Attacks

Attacking the carrier image can be malicious and non-malicious. Malicious attacks occur when the attacker objective is to affect the process of message extraction by tampering the carrier image. For example, the carrier image is attacked by adding noise or cropping. The attack can be also non-malicious, this means the attacker is using the image and change it with regular image processing operations. For example, the image can be rotated, scaled or compressed for transferring and using online.

The attack can be malicious and non-malicious in the same time. For example, the image can be compressed for online share or compressed for the purpose of removing the watermark. In this section, we divide the image attacks to active and passive attacks. In active attacks, the carrier image is tampered in order to remove the watermark or make the extraction process difficult [12].

In passive attacks, the objective is to read the message. Passive attacks are more applied to steganography techniques. Steganalysis is passive attack type that extracts the message or make a binary decision for an image whether it has secret data or not [11].

A. Active attacks

Image enhancement increases the image quality. However, it may remove the embedded message. Image enhancements such as image sharpening, low/high pass filters and histogram equalization remove the entire message embedded with simple LSB procedure [4]. In addition, geometric attacks are more destructive compared to image enhancement attacks. The carrier image can be cropped, scaled, or rotated. The hidden message is affected directly by such attacks whether it is embedded in the frequency domain or in the spatial domain. In the frequency domain, DWT, DCT transforms are applied on the image to get the corresponding coefficients. The watermark is embedded into DCT or DWT coefficients. The problem with this type of embedding is obtaining perfect synchronization in the extraction process. To overcome Rotation, Scaling and Translation (RST) attacks, global features such as (SIFT) features are extracted from the image and used for embedding [20]. In the spatial domain, the amount of embedded data linearly affects the image quality. Most approaches tend to reduce the message length and embed it into the image regions with high intensity fluctuations [21]. Here we divide the carrier image attacks into image enhancement, image degradation and geometric transformation attacks.

A.1. Image Degradation Attacks

The degradation for the image maybe done malicious or non-malicious. The signal maybe degraded with noise interference when transferring it from the sender to the receiver. The attacker degrades the image by adding noise such as "salt and pepper" noise.

Noise Addition. The noise is introduced by imaging system or by an attacker. There are four models for modeling noise, the simple model, the additive/multiplicative noise model, the Gaussian noise model and the impulse noise model [22]. In the simple model, noise at each pixel is independent, the noise is characterized by mean and the corresponding standard deviation for each pixel. Simple model is expressed mathematically as

$$\bar{I}(i,j) = \frac{1}{N}\sum_{k=0}^{N-1} I_k(i,j) \qquad (1)$$

$$\sigma(i,j) = \sqrt{\frac{1}{N-1}\sum_{k=0}^{N-1}(I_k(i,j) - \bar{I}(i,j))^2} \qquad (2)$$

Where N is the number of pixels in the image I_k. The image $\bar{I}$ is the noised image and σ is the standard deviation.

The second model is Additive noise model. In Additive noise model, random noise is added to each pixel. The noise addition can be expressed by

$$I(i,j) + n(i,j) \tag{3}$$

where *I(i,j)* is the original image pixel, *n(i,j)* is the noise added to the original pixel. and $\hat{I}(i,j)$ is the resulted noised image. The image quality can be measured by Signal to Noise Ratio (SNR). The multiplicative noise has the same concept of additive noise but with addition "+" replaced with multiplication "*" in equation (3).

Some noise models maybe additive and subtractive in the same time such as speckle noise model [38]. The generalized speckle noise can be represented as

$$\hat{I}(i,j) = I(i,j) * m(i,j) + n(i,j) \tag{4}$$

Where *m(i, j)* is the multiplicative component and *n(i, j)* is the additive component.

The gaussian model adds random noise to the image with Gaussian noise distribution. If *n(i,j)* is independent and not related to the image data, it is called the White gaussian noise. In other words, *n(i,j)* is noise random variable (depicted as *x* in the following equation) defined by the following distribution

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{x^2}{2\sigma^2}} \tag{5}$$

The last type of noise is Impulse noise. The noise is caused by transmission errors due to interference with other signals. The noise can be caused by an attacker. The impulse, spot or peak noise is also called "salt and pepper" noise. The salt and pepper noise is expressed mathematically as

$$\hat{I}(i,j) = \begin{cases} I(i,j) & x < l \\ I_{min} + y(I_{max} - I_{min}) & x \geq \end{cases} \tag{6}$$

Where I_{min}, I_{max}, are the min and max values of image pixels. x, y are two uniform distributed random binary variables.

Results of attacks on watermarked image with Gaussian noise show extensive destruction of the watermark compared to other noise addition attacks. The watermark can be totally or partially restored when attacked with additive, impulse or Gaussian noise [23]. Sample of gaussian attack on color images is shown in Figure 4 (a) The restoration will be successful only if the watermarking is done locally, i.e. pixel based or block-based watermarking. However, if the data hiding is done globally, the synchronization process might be affected, and the entire watermark won't be restored.

Block Replacement Attack. Blocks in an image could be similar. Block replacement attack (BRA) uses this property in the image to replace some blocks in an image with other blocks in the same image [39]. One straight forward implementation of BRA is to use fractal coding [40].

The input image is firstly partitioned to N non-overlapping blocks called range blocks Ri (the algorithm can be made more accurate if blocks can overlap). For each range block Ri, we search in a window in the image called search block Si. The search block is selected in the vicinity of the range block or selected randomly. Collection of blocks are obtained from within the search range to construct a codebook. Each candidate block in the codebook is compared with Ri. The block with lower Mean Square Error (MSE) is selected as a candidate block for replacement. In fractal coding the codebook is implemented by applying the geometric transforms on the search range sub-blocks. For simplicity, 9 geometric transforms are selected; identity, scale down by a factor of two, 4 flips, and 3 rotations.

Figure 3 shows range and search blocks. For each range block, a search block is created as shown to right side of the figure with big squares. All sub-blocks in the selected search block is added to the corresponding codebook. Each sub block must be rotated, flipped and scaled. All transformed versions of the sub-block are added to the codebook. Finally, for each range block, the codebooks are scanned to find the best match which is candidate for block replacement.

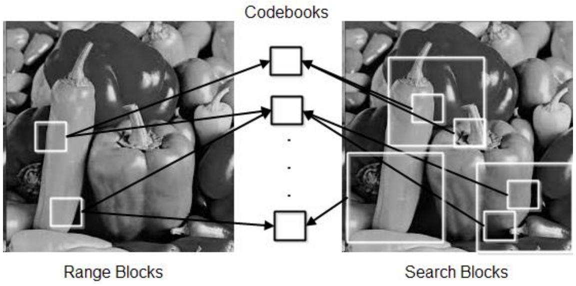
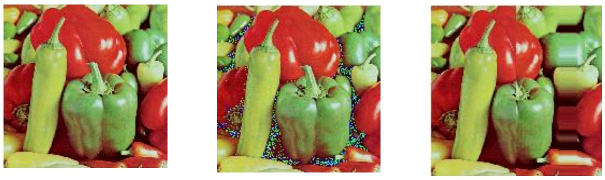


Fig.3. Range blocks represented by white squares on the left image. Search blocks are represented by big squares on the right image, the small squares represent search range sub-blocks.

Jitter Attack. The main purpose of LSB embedding is to find bit locations in low order bits that will not affect the image quality. If the stego-bits locations are defined by a key, the extraction synchronization process will be tampered for simple bits relocation attack. The jitter attack can be simply done with adding jitter to the carrier signal. It adds columns of bits to the image and removes the same number of columns to keep the image size unchanged. The extra columns are copies or interpolated version of neighbor columns. The jitter attack tests the extraction synchronization process [24]. Jitter attack applied on Pepper image is shown in Figure 4 (b)



(a) Original image (b) Noise Addition attack (c) Jitter Attack

Fig.4. Samples of image degradation attacks (a) Original Image (b) Noise addition attack (c) Jitter attack

StirMark Attack. StirMark is implemented mainly to test the capability and robustness of data hiding algorithm [24]. It uses combination of attacks, such as using rotation with shearing or cropping with resizing. It simulates the process of printing the carrier image with high quality printer and scanning it back with high quality scanner. Printing and scanning produces the same error introduced by normal daily usage and locomotion of an image.

StirMark also applies un-noticeable simple geometric distortions. It stretches, rotates, shifts, reflects, mirrors and/or shears the image with small random value that makes the distortion imperceptible. The image is then resampled by bi-linear interpolation. In addition, the attack adds a global random value that distributed uniformly on the entire image. It adds the same effect of the noise created by simple analog to digital and digital to analog converter.

StirMark can be repeated with more than one iteration until the degradation is noticeable. The image watermarking system may survive few iterations, but with more iterations introduced, the watermark maybe completely distorted.

The authors in [24] stated that using linear-interpolation in image reconstruction blurs the image too much. Instead, they used another type of interpolation expressed mathematically as

The authors in [24] stated that using linear-interpolation in image reconstruction blurs the image too much. Instead, they used another type of interpolation expressed mathematically as

$$\hat{I}(i,j) = \sum_{r=-n}^{n}\sum_{k=-n}^{n} \sin c(i-r) \sin c(j-k) f(r,k) \qquad (7)$$

Where $\hat{I}(i,j)$ is the function to be reconstructed. *i,j* are coordinates of the inverse transform. f is the function to reconstructed. *sinc* function is a way to represent waveforms. It represents the waveform but without higher frequency components. When using *sinc* interpolation, no higher harmonics will be added. *sinc* waveform is defined mathematically as

$$sinc(i) = \begin{cases} \frac{\sin(\pi i)}{\pi i}, & if\ i \neq 0 \\ 1 & if\ i = 0 \end{cases} \qquad (8)$$

Data hiding systems are considered useless if they are not robust to StirMark attacks. Authors in their research can overcome specific attacks, they are not able to ensure robustness to all attacks. One might use repeated embedding of the watermark to increase watermark robustness [23]. Other authors tend to protect their hiding technique against rotation and scaling [25].

Random Bending Attack. Stirmark adopts special type of attacks called Random Bending attack. The attack exploits the fact that Human visual system is not sensitive to tiny local attacks. The attack shifts pixels or generally apply affine transforms on local parts of the image. Watermarking systems are not resistant to such attack. but still the embedding algorithm can be manipulated to resist such attack [26]. Random bending is different from StirMark, StriMark is applied globally on the image but random bending is applied locally.

Mosaic Attack. The main objective of this attack is to detect the robustness of the watermarking technique against slicing the image into small sub-images and then reassemble them back. The slicing is done on the server side of a web browser, and the web browser is responsible for displaying the slices side by side to construct an image mosaic. The image mosaic is the same as the image but displayed sliced. The problem with watermarking system that it can't handle embedding in a very small image slice. The embedding capacity will be small and useless. If the embedding is done in small slices, the extraction algorithm will be confused [27]. In some cases, the downloading of mosaic with small chunks is faster than downloading the entire image in one chunk.

Image Compression. Compression means reducing the size of the image, so it can be downloaded easily when published online. The lossy compression removes data which doesn't affect the overall quality of the image. It can reduce the image size to about 5% of its normal size. The problem is that the hidden data may be removed partially or completely due to compression. JPEG compression is mainly used in browsers for progressive download. JPEG compression rearranges image data in such a way that when downloading a small portion of the image data, the overall structure of the image can be precepted. JPEG compression uses Discrete Cosine Transform (DCT) in order to transform the image into frequency domain. However, using DCT in the image compression introduces the blocky and blurry effects. Therefore, JPEG 2000 compression uses Discrete Wavelet Transform (DWT) which overcome the blocky and blurry artifacts and achieved higher compression ratio [17]. After embedding the watermark, artists use the image by applying some enhancements and then save it in a compressed format. Data hiding technique is considered useless if it can't withstand compression.

A.2. Image Enhancement Attacks

The purpose of enhancement is to remove noise from the image or generally make the image as clear as possible. Filtering and histogram equalization are two examples of image enhancement.

Filtering. Image filtering can be done in the frequency domain or in the spatial domain [13]. Examples of image enhancement in the spatial domain are median and mean filters. Median filter aims to remove the impulse noise. Median filtering is done by dividing the image into overlapped blocks, then the median value of the block is calculated and inserted into the center pixel. The same procedure is applied to the mean filter. The mean value is calculated for the block and inserted into the center pixel. Median filter removes spikes such as salt and pepper noise, while mean filter smooths sharp edges in the image.

Image filtering can be done in the frequency domain.

Discrete Fourier Transform (DFT) is calculated for the image in order to transform image data into the frequency domain. The result is then multiplied by the filtering transfer function. Finally, the Inverse Fourier Transform (IDFT) is calculated to obtain the filtered original image. The image filtering can be expressed mathematically [13] as:

$$\mathrm{g}(\mathrm{x}, \mathrm{y}) = \mathrm{IDFT}[H(u,v) \times F(u,v)] \tag{9}$$

Where g is the filtered image in the spatial coordinates x, y. F is the Fourier transform of the original image which is depicted in the transform domain with the coordinates u, v. H is the filter transfer function. H is also depicted in the transform domain. *Sharpening filter* is an example of frequency domain high-pass filters [13]. Sharpening increases the high frequency components and decreases the low frequency components. Gaussian transfer function is used for image sharpening. The Gaussian high-pass transfer function is depicted as:

$$H(u,v) = e^{-\frac{D^2(u,v)}{2\sigma^2}} \tag{10}$$

Where *D(u, v)* is distance between the center of the frequency rectangle and a point (u, v). σ is the standard deviation of the frequencies. If the secure data is hidden in the low frequency component, it might be severely degraded. However, sharpening filter can help in detecting the high frequency components caused by the data hiding techniques [18].

Blurring filter is an example of low-pass filters. The sharp edges in the image are soothed. The filter keeps low frequency components and removes high frequency components. The Gaussian low-pass transfer function is typically used for image blurring. It is depicted as:

$$H(u,v) = 1 - e^{-\frac{D^2(u,v)}{2\sigma^2}} \tag{11}$$

If the hidden data exists in the high frequency components - which is typically the case in many hiding algorithms [21], the hidden data might be removed in the filtering process. Figure 5 shows an example of image filtering using blurring and sharpening filters.

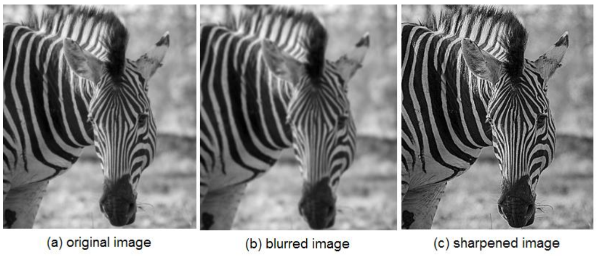


Fig.5. Image filtering (a) original image (b) blurred image, most of high frequency components are removed (c) sharpened image, high frequency components are intense and focused.

Histogram Modification. Histogram modification changes the color frequency either by stretching or equalization. Histogram equalization improves image lightening [13].

Suppose we have discrete gray levels or image colors. The input gray levels can take the following form

$$X=[\, x_0, x_0, x_1, x_1, x_1, x_3, x_{10}, \ldots\ldots.., x_{L-1}, x_{L-1}], \tag{12}$$

X is a vector with N elements

Where *X* represents the input image, and x’s are the gray scales. Grayscale 0 for example is repeated twice, grayscale 1 is repeated 3 times and so on. L is the number of grayscale levels which is typically 255 in gray images. By using the total number of pixels (N), vector X can be used to construct the histogram. However, to make histogram equalization, both Probability Distribution Function (PDF) and Cumulative Distribution Function (CDF) should be calculated. PDF is calculated by counting the occurrences of each gray level then divide number of occurrences by N. For example, x_0 in equation (12) occurred 2 times. So its probability = 2/N. x_1 occurred 3 times, so its probability = 3/N. The same procedure is applied to calculate all probabilities for the remaining x values. CDF is calculated by summing up until the current x. For example, at x_0 we have CDF=2/N. At x_1, we have CDF=2/N + 3/N. At x_3, we have CDF=2/N+3/N+1/N. The same procedure is repeated until reaching x_L, at x_L, CDF = 1. The following equations represent the above procedures

$$PDF(x_i) = \frac{n_i}{N} \tag{13}$$

$$CDF(x_j) = \sum_{i=0}^{i} PDF(x_i) \tag{14}$$

Where n_i is the number of occurrences of intensity level i. The new intensity levels k created by histogram equalization can be calculated as:

$$k_j = \lfloor (L-1) \times CDF(x_j) + 0.5 \rfloor \tag{15}$$

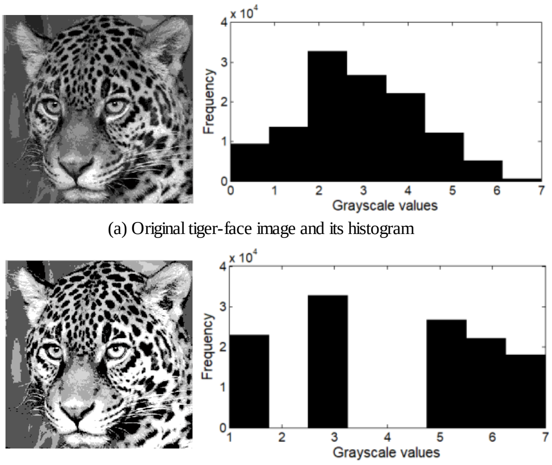
The value 0.5 is added for proper flooring. Notice that the equalization equation above maps CDF of maximum gray scale to *L-1*, and the minimum grayscale – which is 0, to 0. So, the entire span of gray levels is included in the equalized grayscales [31]

Let us take an example. The tiger-face image shown in Figure 6 is 350 x 350 pixels grayscale image. For simplicity, we have quantized the image, so it contains only 8 grayscale levels. The corresponding histogram is calculated, the tiger-face image and its corresponding histogram are shown in Figure 6 (a).

The frequencies are 9342, 13669, 32751, 26706, 22046, 12153, 5182, and 651 for gray levels 0, 1 ,2, 3, 4, 5, 6, and 7 respectively. The bar heights of the histogram in Figure 6 (a) represent pixel frequencies, the horizontal axis represents grayscales.

The corresponding probabilities (PDF) calculated by equation (13) are (0.076, 0.111, 0.267, 0.218, 0.179, 0.099, 0.042, 0.008). As an example, PDF for grayscale 0 is calculated as PDF (0) = 9342/122500=0.076.

The corresponding cumulative probabilities (CDF) calculated by equation (14) are (0.076, 0.187, 0.455, 0.673, 0.853, 0.952, 0.994, 1). As an example, CDF (1) = PDF (0) + PDF (1) = 0.076+0.111=0.187.



(a) Original tiger-face image and its histogram

(b) Histogram equalized tiger-face image and its histogram

Fig.6. Histogram equalization (a) Original tiger image quantized into 8 grayscale levels, to the right, its corresponding histogram. (b) The histogram equalized version of tiger-face image showing how the frequencies of the histogram are redistributed to cover all grayscale span.

The equalized grayscales are calculated by multiplying the grayscale's CDF with L-1 and then flooring the result, equation (15). For example, for grayscale 5, the new grayscale is floor ((8-1) * CDF (5) + 0.5) =floor (7*0.952+0.5) =7. So, image pixels with grayscale 5 will be changed to grayscale 7. Using equation (15) all the remaining grayscales can be calculated. The resulting grayscales are (1, 1, 3, 5, 6, 7, 7, 7).

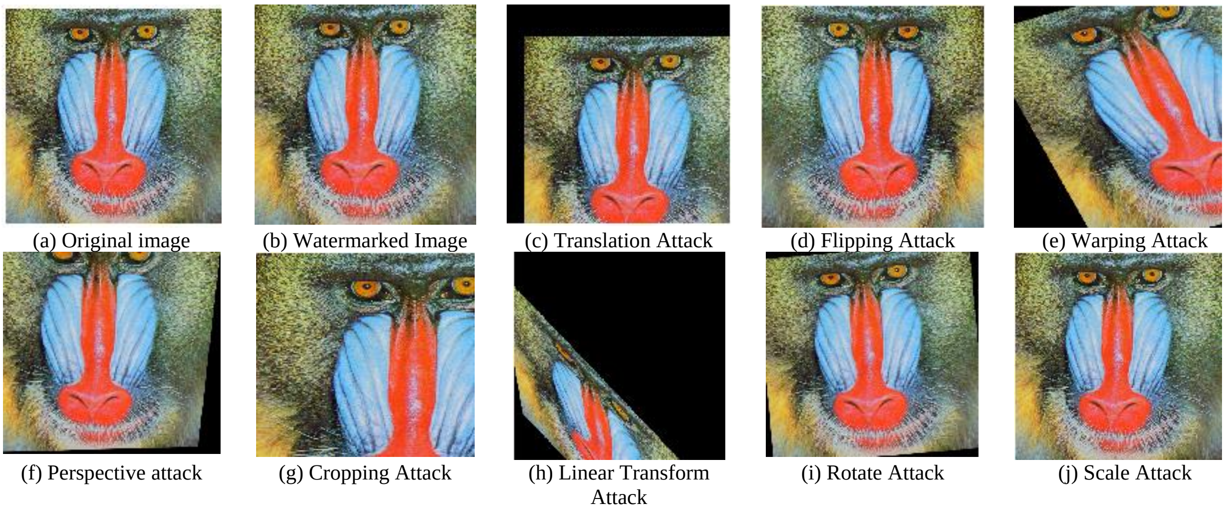
Figure 6(b) shows the equalized image with its corresponding histogram. Notice that, the two histograms in Figure 6 differs in distribution type. The original image histogram has nearly bell shaped which is interpreted as normal distribution. After equalization, the resulting histogram frequencies have nearly similar values. That is interpreted as uniform distribution. Uniform distribution means giving approximately equal probabilities for all grayscales.

The embedded data is affected by histogram equalization. Grayscale values are changed which result in perturbing the embedded data. If the hidden data is embedded in the frequency domain, it will be affected by histogram equalization because the equalization process increases high frequency components.

A.3. Geometric Transforms

Geometric change means changing the overall shape of the image. Transforms such as, rotation, scaling and cropping changes image grayscale values by adding or removing pixels. Geometric transform based attacks are shown in Figure 7. This can affect directly the embedded data. Geometric transformations are generally divided into affine transformations and projective transformations. Affine transformations keep parallel lines parallel and keeps angle values without change; it may change the aspect ratio. Examples of affine transformations are rotation, translation, scaling and cropping. Projective transformations transform lines to lines while not essentially keeps parallelism and angle values. Examples of projective transformations are perspective transform, shearing and warping. Affine transform can be expressed as 3x3 invertible matrix while projective transforms can be expressed as 3x3 non-invertible matrix. In affine transform, scaling and rotation uses 2x2 matrix as a 2D transform. In translation transform, a 2x2 matrix can't be used. Instead 3x3 homogenous coordinates matrix is used. The method in [37] is robust to most geometric attacks.

Cropping. After taking photos or snapshots, artists are only interested in specific part of the snapshot. Artists often apply cropping after document scanning or taking snapshots. The image maybe downloaded from the internet and used partially. If the image has hidden data, the hidden data inside the cropped image will be affected.



(a) Original image (b) Watermarked Image (c) Translation Attack (d) Flipping Attack (e) Warping Attack

(f) Perspective attack (g) Cropping Attack (h) Linear Transform Attack (i) Rotate Attack (j) Scale Attack

Fig.7. Geometric Transform Attacks.

Data hiding approaches can be implemented to avoid cropping. A typical approach is to divide the image into non-overlapping blocks, and then embed a copy of the watermark in each block [14, 15]. If the image is cropped version, in some cases it is important to get the original image and then return the cropped part to the original image so as to be able to extract part of the watermark [17].

In case of blind watermarking – where the original image is not used in the watermarking, the most robust approach to clipping attack is to distribute copies of the watermark on the image. Sudoku game grid can be utilized to reconstruct the watermark from cropped watermarked image [28], but still the watermarking is non-blind, and the Sudoku grid must be accompanied with the watermarked image during the extraction process. It is better to avoid cropping attack by embedding only in regions of interest (RoI).

Flipping. Flipping an image means getting the mirror of the image. It can be done horizontally or vertically. The problem with this attack is that it changes the pixel positions. When extracting the hidden data with spatial domain technique, synchronization will be lost, and the watermark can't be extracted.

With frequency domain watermarking, frequencies remain the same. If the embedding is done on the frequencies of the entire image, hidden data can be restored efficiently. But if the embedding procedure divides the image into blocks before calculating each block's frequencies, the embedded data will be lost due to synchronization problem. If single bit is embedded in each block, a flipped version of the watermark can be extracted. However, flipping attack can be avoided simply by implementing an embedding procedure resilient to image flipping attack [5]. Flipping can be expressed in matrix form as a mirror or reflection transform:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{16}$$

The reflection is done through x-axis, the reflection can be done also on y-axis or x-y axis. The operation converts discrete coordinates into negative discrete coordinates. Originally, the image has no negative coordinates. To overcome this problem, the coordinate axis must be translated to new location. In other words, all the coordinates must be recalculated according to the new origin.

Rotation. After scanning an image, the image may need small angle rotation combined with cropping. This may not affect the overall shape of the image, but the watermark maybe partially or completely removed [32]. Rotation is needed to align objects in the image vertically or horizontally. Rotation is 2D transform which maps point with coordinates (x, y) to a new location with coordinates (x', y'). For 2D geometry – which is defined by vertex coordinates with floating point numbers, rotation is achieved simply by changing these coordinates [29]. However, in an image, pixel coordinates are changed to new coordinates which may not be exist. This occurs because of two reasons: First, the image coordinates are discrete, and the calculated coordinates are continuous. A method must be found to do reasonable rounding such as interpolation. Second, the new coordinates maybe located outside the image border. A method for proper cropping is needed after rotation.

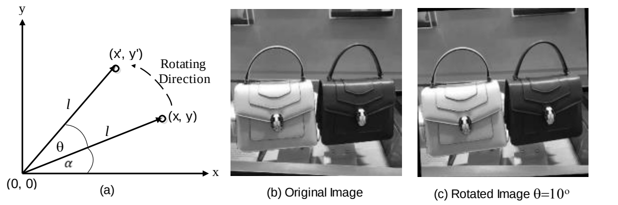


Fig.8. Rotation of an image, (a) the rotation angles (b) original image (c) the image after applying rotation 10o counter clock-wise.

Suppose there is an image pixel with coordinates *(x, y)* need to be rotated θ degrees counter-clockwise around origin as shown in Figure 8 (a). After rotation the new coordinates of the pixel will be (x', y'). To calculate x', y', the following equations are applied:

$$x = l\cos(\alpha),\ y = l\sin(\alpha) \tag{17}$$

$$x' = l\cos(\alpha + \theta),\ y' = l\sin(\alpha + \theta) \tag{18}$$

where *l* is the distance between the pixel position and the origin. From trigonometry

$$\begin{aligned} x' &= l\cos(\alpha)\cos(\theta) - l\sin(\alpha)\sin(\theta), \\ y' &= l\cos(\alpha)\sin(\theta) + l\sin(\alpha)\cos(\theta) \end{aligned} \tag{19}$$

substituting (17) in (19)

$$x' = x\cos(\theta) - y\sin(\theta),\ y' = x\sin(\theta) + y\cos(\theta), \tag{20}$$

The above equations can be expressed in matrix form as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{21}$$

Figure 8 (b), (c) shows an image and its rotated version. The above equations are used to rotate the image 10^o counter-clockwise. The image pixels with black color in the borders of the rotated version are not exist in the rotated coordinates set. It is substituted by black color. In addition, parts of the image are mapped to locations outside the image border. These parts are cropped. For data hiding systems, three types of perturbation must be considered.

The first perturbation occurred due to the non-mapped pixels, which are depicted in black color. The second perturbation happens due to the pixels that mapped outside the image borders. The third perturbation is the change of the entire image pixel coordinates. Image rotation is very destructive to the embedded data. It must be considered in the implementation phase of the data hiding algorithm.

Scaling. Scaling is the process of enlarging or reducing the image dimensions. It means adding rows and columns to the image when scaling up; and removing rows and columns of the image when scaling down. The method of bilinear interpolation is adopted for keeping the image display quality and keeping smooth appearance [33].

Bicubic interpolation can also be adopted to provide higher quality of scaled image. Adjusting images size is important in web publishing. Scaling can be uniform and non-uniform. In uniform scaling, the scaling is done equally in both horizonal and vertical directions. In non-uniform scaling, different scaling factors are used in horizontal and vertical directions.

Non-uniform scaling changes the aspect ratio of the image. It can be verified that the data hiding approaches are more robust to uniform scaling than non-uniform scaling. In both approached, data is removed and/or added to the image. This affects directly the embedded data by exposing it to distortion. Scaling can be expressed in matrix format as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{22}$$

Where S_x is the scale in horizontal direction and S_y is the scale in vertical direction. If the value 1 is assigned to S_x or S_y, then no scale is applied to the corresponding direction. Values greater than 1 scale up the image in the related direction, and values less than 1 scale down the image in the related direction. In uniform scaling, $S_x = S_y$ and in non-uniform scaling, $S_x \neq S_y$.

Shearing. Shearing is an attack that slants the image shape. Little shearing can be acceptable and used in some applications [34]. Shearing is also called skewing transform. The watermarked image is directly affected by shearing transform. There are three types of shearing; x-shearing and y-shearing and x-y shearing. Shearing can be expressed in matrix form as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & Sh_y \\ Sh_x & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{23}$$

Linear Transformation. It is a general, the linear transform is composed of other transforms. For example, the carrier image can be rotated, scaled and sheared [35]. Liner transform can do all three transformations in one composite transform. This is accomplished by taking all the corresponding transform matrices and multiply them together. The resulting matrix is used for the linear transform. Suppose it is needed to rotate the image 10^o counter-clockwise and scale it 2,2 in both direction and shear it 1.5 in x. The resulting multiplication should be calculated as:

$$\begin{aligned} \begin{bmatrix} x' \\ y' \end{bmatrix} &= \begin{bmatrix} \cos(10) & -\sin(10) \\ \sin(10) & \cos(10) \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} -0.98 & -1.97 \\ 3.30 & 1.97 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned} \tag{24}$$

Warping Attack. Image Warping takes an input image and convert it to another image reshaped according to specific mapping or transform. A texture image in a square shape can be warped to cover a tilted floor. Warping is often used in image mosaic to create panoramic image. Warping idea is to take each pixel of the image and apply the mapping transform to get new location of the pixel. The pixel color and the number of channels remains the same. Some pixels are removed from the image due to the reshaping process. If the image is stego-image, then hidden data is affected directly by warping.

B. Passive Attacks

Passive attacks aim to discover the secret message. Such attacks are more suitable for steganography techniques when knowing the secure message is a vital task.

B.1. Visual Attack

Visual attack is the only human attack type. It depends on Human Visual System (HVS) to detect the existence of hidden data inside the carrier image [30]. Embedding in the least significant bit (LSB) of the image changes the randomness of LSB bits. Unfortunately, after embedding, machines can’t discriminate between the original LSB randomness and the carrier LSB randomness. The task of differentiation is harder if the algorithm embeds data into specific random pixels in the image. For example, authors in [21] embeds data into the image regions with abrupt changes. For humans, they still can differentiate image/stego-image with their extensively complex visual system. Humans are trained on recognizing objects even in random environments – Who didn’t recognize shapes in sky clouds? [30].

A simple example for visual attack is shown in Figure 9. Image planes are used as a carrier signal for the embedding of secure data. All LSB’s of the image are used for embedding secure data. Each pixel in the image is represented by 8 bits number. Figure 9 (a) shows the results of embedding the secure data in single LSB. Figure 9(b) shows the results of embedding the data in 2 LSBs.

Figure 9 (c) shows the results of embedding the secure data in 4 LSBs. Figure 9 (b) shows the results of embedding the secure data in 6 LSBs. The lower images are the first LSB bit plane. The bit planes show that LSB are naturally random, machine finds difficulty to discriminate between the stego-bit-planes and clear bit planes. Even with surge perturbation which are easily detected with HVS as shown in Figure 9 (d), bit-planes are still random and hard to discriminate.

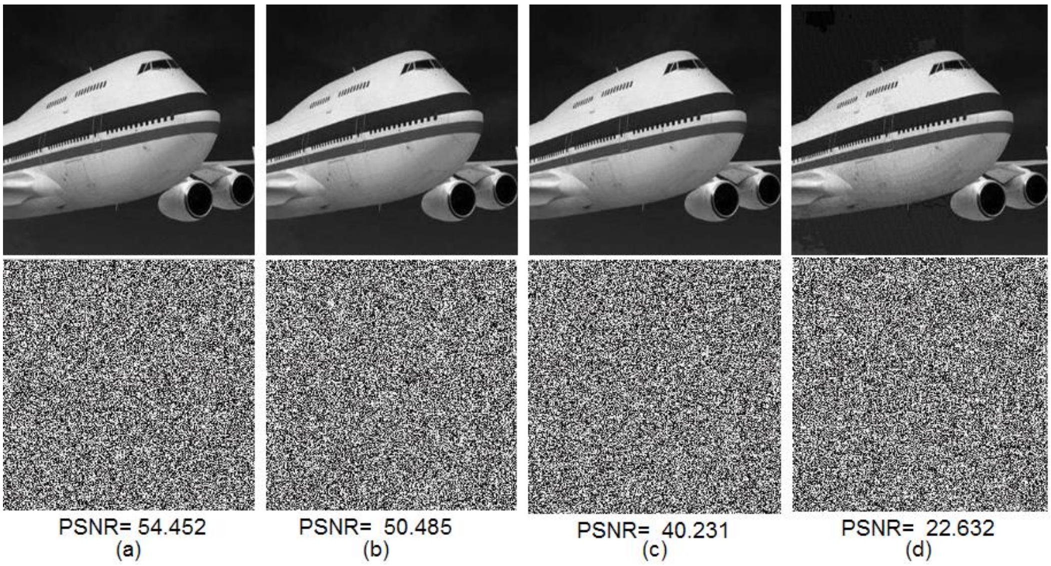


Fig.9. Plane image after embedding secure data in all LSB's of the image (a) stego-image with the first LSB bit is used for embedding (b) stego-image with 2 LSB's are used for embedding (c) stego-image with 4 LSB's are used for embedding (d) stego-image with 6 bits are used for embedding. The lower images are the first LSB bit plane representing how machines can't discriminate randomness but human visual system (HVS) can discriminate easily. It is appearing in stego-image in (d) some embedding artifacts appear on top of the plane.

B.2. Chi-Square attack

This attack was originally developed by Westfeld and Pfitzmann [30]. First, chi-squared analysis is a statistical test to measure the similarity of two sets of data. The first dataset is called the observed dataset and the second dataset is the expected dataset. The main idea of chi-square attack is to compare Pair of Value (PoV) of the observed data and compare it with the expected data and accordingly calculate the probability of having embedded data in the image.

PoVs refers to the frequencies of the pair of pixel values in the image. For example, there are 128 pairs in 8-bits grayscale image. The pairs are {(0, 1), (2, 3), (4, 5) , ……. , (254, 255)}. The pairs are represented in binary as {(00000000, 00000001), (00000010, 00000011), (00000100, 00000101) , ……. , (11111110, 11111111)}. The authors in [21] discovered that, when embedding random encrypted data into LSB of the image, PoVs frequencies are approximately equal. While in clear images, PoV frequencies differ so much.

Chi-square test is applied by calculating the frequencies of stego-image color values. Therefore, the observed dataset is obtained without using the original image. The expected values can be obtained by taking the average of the expected PoV frequencies. The observed average is used as the expected average because the frequencies of PoV will differ in values but similar in mean. The increase in one frequency value is deducted from another frequency value. PoV's are divided into n categories which is typically 128 in our discussion. The statistic Chi-square (X^2) is given mathematically as

$$X^2_{n-1} = \sum_{i=1}^{n} \frac{(D_{oi} - D_{ei})^2}{D_{ei}} \quad (25)$$

D_{oi} is the observed frequency distribution of PoV. D_{ei} is the expected frequency distribution of PoV taken as the average of the frequency values of PoV.

The probability of obtaining suspected data embedded into the observed image is measured by p-value. It is represented as

$$p = 1 - \frac{1}{2^{\frac{n-1}{2}}\Gamma(\frac{n-1}{2})} \int_0^{X^2_{n-1}} e^{-\frac{x}{2}} x^{\frac{n-1}{2}-1} dx \quad (26)$$

P value is the probability that the image has embedded data. As a result the attack can detect in what percentage the image is suspect to have embedded data.

B.3. AI based attacks

Artificial intelligence (AI) is the ability of the machine to resemble human behavior. Rules are added to the intelligent agent to be able to "Think" like human. Furthermore, Machine learning is the area of AI in which rules are not fully added to the agent, instead new rules can be inferred by the learning process. In data hiding, the embedding process leaves traces which can be easily detected by statistical methods [42]. Syndrom-trellis codes are special codes used to reduce the embedding distortion results from adding data to the original image. However, machine learning based approaches such as Artificial Neural Network (ANN) [43] can detect the existence of embedded data. Machine learning based models such as rich models [44] and deep learning models [45, 49].

Deep Learning models are more accurate in detecting hidden data. Deep learning uses Neural Networks with large number of hidden layers (deep layers). The deep learning process contains two phases. The first phase is the learning process, in which hundreds of labeled images are passed to the model for training. The images are labeled as either clear or suspect. The more the provided learning dataset, the more accurate the model. The training dataset is usually enlarged, and attention is payed to avoid the overfitting problem. The second phase is the test process, in which a suspect image is passed to the deep model to check the existence of hidden data. Convolutional Neural Network model shows efficiency in detecting the embedded data [46]. Research scientists proposed an approach called steganography without embedding (SWE) [47, 48] for avoiding the detection by deep learning models. SWE embeds data without perturbing the carrier image.

IV. Conclusion

The paper presented different types of attacks that exploit vulnerabilities in data hiding techniques using an image as a carrier signal. We started by presenting secret message attacks such as Oracle, desynchronization and template attacks. Passive and active attacks which are directed to the carrier image are discussed. Most of the attacks presented are active attacks. Active attacks are detailed and separately discussed. Some of active attacks are malicious and others are non-malicious. Active attacks also can be accomplished with image enhancement or image degradation techniques. A very destructive type of active attacks is geometric attack specially the projective type. The projective type changes the angle and parallelism of the image content while affine attack type keeps parallelism and angle values.

References

[1] Cox, I., Miller, M., Bloom, J., Fridrich, J., & Kalker, T. (2007). *Digital watermarking and steganography*. Morgan Kaufmann.

[2] Liu, S., Pan, Z., & Song, H. (2017). Digital image watermarking method based on DCT and fractal encoding. *IET Image Processing*, *11*(10), 815-821.

[3] Brandao, A. S., & Jorge, D. C. (2016). Artificial neural networks applied to image steganography. *IEEE Latin America Transactions*, *14*(3), 1361-1366.

[4] Fridrich, J., Goljan, M., & Du, R. (2001). Detecting LSB steganography in color, and gray-scale images. *IEEE multimedia*, *8*(4), 22-28.

[5] Voloshynovskiy, S., Pereira, S., Pun, T., Eggers, J. J., & Su, J. K. (2001). Attacks on digital watermarks: classification, estimation based attacks, and benchmarks. *IEEE communications Magazine*, *39*(8), 118-126.

[6] Hosam, O. (2013). Side-informed image watermarking scheme based on dither modulation in the frequency domain. *The Open Signal Processing Journal*, *5*(1), 1-6.

[7] Linnartz, J. P. M., & Van Dijk, M. (1998, April). Analysis of the sensitivity attack against electronic watermarks in images. In *International Workshop on Information Hiding* (pp. 258-272). Springer, Berlin, Heidelberg.

[8] S. Craver et al., "On the Invertibility of Invisible Watermarking Techniques," Proc. IEEE Int'l. Conf, Image Processing 1997, vol. 1, pp. 540-43.

[9] Memon N, Shende S, Wong PW. On the security of the Yeung-Mintzer authentication watermark. Proceedings of the IS & T PICS Symposium; March 1999; Savannah, Ga, USA. pp. 301–306

[10] Tao, H., Chongmin, L., Zain, J. M., & Abdalla, A. N. (2014). Robust image watermarking theories and techniques: A review. *Journal of applied research and technology*, *12*(1), 122-138.

[11] Boroumand, M., & Fridrich, J. (2018). Applications of Explicit Non-Linear Feature Maps in Steganalysis. *IEEE Transactions on Information Forensics and Security VOL. 13, NO. 4.*

[12] Sujatha, C. N., & Satyanarayana, P. (2016, April). Analysis of robust watermarking techniques: A retrospective. In *Communication and Signal Processing (ICCSP), 2016 International Conference on* (pp. 0336-0341). IEEE.

[13] Gonzalez, R. C., & Woods, R. E. (2007). Image processing. *Digital image processing*, *2*.

[14] Khalid, S. K. A., Deris, M. M., & Mohamad, K. M. (2013). Anti-cropping digital image watermarking using Sudoku. *International Journal of Grid and Utility Computing*, *4*(2-3), 169-177.

[15] Hosam, O., & Alraddadi, A. S. (2013). Novel image watermarking technique based on adjacent pixel position switch. *Journal of Next Generation Information Technology*, *4*(3), 81.

[16] Preda, R. O., & Vizireanu, D. N. (2015). Watermarking-based image authentication robust to JPEG compression. *Electronics Letters*, *51*(23), 1873-1875.

[17] Shih, F. Y. (2017). *Digital watermarking and steganography: fundamentals and techniques*. second edition, CRC Press.

[18] Kutter, M., & Petitcolas, F. A. (1999). A Fair benchmark for image watermarking systems. *Security and Watermarking of Multimedia Contents*, *3657*, 226-239.

[19] Chang, Y., & Tai, W. (2013). A block-based watermarking scheme for image tamper detection and self-recovery. *Opto-Electronics Review*, *21*(2), 182-190.

[20] Lin, Y. T., Huang, C. Y., & Lee, G. C. (2011). Rotation, scaling, and translation resilient watermarking for images. *IET Image Processing*, *5*(4), 328-340.

[21] Hosam, O., & Ben Halima, N. (2016). Adaptive block‐based pixel value differencing steganography. *Security and Communication Networks*, *9*(18), 5036-5050.

[22] Ponce, J., Forsyth, D., Willow, E. P., Antipolis-Méditerranée, S., d'activité RAweb, R., Inria, L., & Alumni, I. (2011). Computer vision: a modern approach. *Computer*, *16*(11).

[23] Hosam, O., & Alraddadi, A. S. (2013). Novel image watermarking technique based on adjacent pixel position switch. *Journal of Next Generation Information Technology*, *4*(3), 81.

[24] Petitcolas, F. A., Anderson, R. J., & Kuhn, M. G. (1998, April). Attacks on copyright marking systems. In International *workshop on information hiding* (pp. 218-238). Springer, Berlin, Heidelberg.

[25] O'Ruanaidh, J. J., & Pereira, S. (1998, September). Secure robust digital image watermark. In *Electronic Imaging: Processing, Printing, and Publishing in Color* (Vol. 3409, pp. 150-164). International Society for Optics and Photonics.

[26] Licks, V., & Jordan, R. (2005). Geometric attacks on image watermarking systems. *IEEE multimedia*, *12*(3), 68-78.

[27] Petitcolas, F. A. (1997). Weakness of existing watermarking schemes, http://www.cl.cam.ac.uk/~fapp2/watermarking /image_watermarking/

[28] Behravan, B., & Naghsh, A. (2017, April). Introducing a new method of image reconstruction against crop attack using sudoku watermarking algorithm. In *Pattern Recognition and Image Analysis (IPRIA), 2017 3rd International Conference on*(pp. 177-181). IEEE.

[29] Hearn, D. D., Baker, M. P., & Carithers, W. (2010).

Computer graphics with open GL. Prentice Hall Press.

[30] Westfeld, A., & Pfitzmann, A. (1999, September). Attacks on steganographic systems. In *International workshop on information hiding* (pp. 61-76). Springer, Berlin, Heidelberg.

[31] Y. Chang, C. Jung, P. Ke, H. Song and J. Hwang, "Automatic Contrast-Limited Adaptive Histogram Equalization With Dual Gamma Correction," in *IEEE Access*, vol. 6, pp. 11782-11792, 2018. doi: 10.1109/ACCESS.2018.2797872

[32] P. Liu and Y. Q. Jin, "A Study of Ship Rotation Effects on SAR Image," in IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 6, pp. 3132-3144, June 2017. doi: 10.1109/TGRS.2017.2662038

[33] Q. Su, "Novel blind colour image watermarking technique using Hessenberg decomposition," in IET Image Processing, vol. 10, no. 11, pp. 817-829, 11 2016. doi: 10.1049/iet-ipr.2016.0048

[34] A. Girdhar and V. Kumar, "Comprehensive survey of 3D image steganography techniques," in IET Image Processing, vol. 12, no. 1, pp. 1-10, 1 2018. doi: 10.1049/iet-ipr.2017.0162

[35] Chih-Wei Tang and Hsueh-Ming Hang, "A feature-based robust digital image watermarking scheme," in IEEE Transactions on Signal Processing, vol. 51, no. 4, pp. 950-959, Apr 2003. doi: 10.1109/TSP.2003.809367

[36] Juarez-Sandoval, O., Fragoso-Navarro, E., Cedillo-Hernandez, M., Cedillo-Hernandez, A., Nakano, M., & Perez-Meana, H. (2018). improved imperceptible visible watermarking algorithm for auxiliary information delivery. *IET Biometrics*.

[37] Shahdoosti, H. R., & Salehi, M. (2017). Transform-based watermarking algorithm maintaining perceptual transparency. *IET Image Processing*, *12*(5), 751-759.

[38] Patiño, A., Altamar, H., & Martínez-Santos, J. C. (2016, August). Speckle free optical watermarking based on pseudo-random phase encoding. In *Signal Processing, Images and Artificial Vision (STSIVA), 2016 XXI Symposium on* (pp. 1-6). IEEE.

[39] Doërr, G., Dugelay, J.L. "Countermeasures for collusion attacks exploiting host signal redundancy" Int. Workshop on Digital Watermarking, 2005, pp.216–230

[40] Y. Fisher, Fractal Image Compression: Theory and Application, editor Springer-Verlag, New York, 1995.

[41] Zhang, Xinpeng, and Shuozhong Wang. "Watermarking scheme capable of resisting sensitivity attack." *IEEE Signal Processing Letters* 14.2 (2007): 125-128.

[42] C. Cachin, ''An information-theoretic model for steganography,'' in Information Hiding. Berlin, Germany: Springer, 1998, pp. 306–318. [Online]. Available: https://doi.org/10.1007/3-540-49380-8_21

[43] Brandao, A. S., & Jorge, D. C. (2016). Artificial neural networks applied to image steganography. IEEE Latin America Transactions, 14(3), 1361-1366.

[44] M. Goljan, J. Fridrich, and R. Cogranne, ''Rich model for steganalysis of color images,'' in Proc. IEEE Int. Workshop Inf. Forensics Secur., Dec. 2015, pp. 185–190

[45] J. Zeng, S. Tan, B. Li, and J. Huang. (Nov. 2016). ''Large-scale JPEG steganalysis using hybrid deep-learning framework.'' [Online]. Available: https://arxiv.org/abs/1611.03233

[46] Ye, Jian, Jiangqun Ni, and Yang Yi. "Deep learning hierarchical representations for image steganalysis." IEEE Transactions on Information Forensics and Security 12.11 (2017): 2545-2557.

[47] M. Barni, ''Steganography in digital media: Principles, algorithms, and applications (Fridrich, J. 2010) [book reviews],'' IEEE Signal Process. Mag., vol. 28, no. 5, pp. 142–144, Sep. 2011.

[48] Hu, D., Wang, L., Jiang, W., Zheng, S., & Li, B. (2018). A Novel Image Steganography Method via Deep Convolutional Generative Adversarial Networks. IEEE Access, 6, 38303-38314.

[49] Wu, Songtao, Shenghua Zhong, and Yan Liu. "Deep residual learning for image steganalysis." Multimedia tools and applications 77.9 (2018): 10437-10453.

[50] Johnson, Neil F., Zoran Duric, and Sushil Jajodia. Information Hiding: Steganography and Watermarking-Attacks and Countermeasures: Steganography and Watermarking: Attacks and Countermeasures. Vol. 1. Springer Science & Business Media, 2001.

[51] Tanha, M., Torshizi, S. D. S., Abdullah, M. T., & Hashim, F. (2012, June). An overview of attacks against digital watermarking and their respective countermeasures. In Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on (pp. 265-270). IEEE.

[52] Wang, Huaiqing, and Shuozhong Wang. "Cyber warfare: steganography vs. steganalysis." Communications of the ACM 47.10 (2004): 76-82.

[53] Pérez-Freire, L., Comesana, P., Troncoso-Pastoriza, J. R., & Pérez-González, F. (2006). Watermarking security: a survey. In Transactions on Data Hiding and Multimedia Security I (pp. 41-72). Springer, Berlin, Heidelberg.

[54] Bas, P., & Westfeld, A. (2009, September). Two key estimation techniques for the broken arrows watermarking scheme. In Proceedings of the 11th ACM workshop on Multimedia and security (pp. 1-8). ACM.

[55] Mauro Barni, Franco Bartolini, Teddy Furon. A general framework for robust watermarking security. Signal Processing, Elsevier, 2003, 83 (10), pp.2069- 084.

[56] Authors: Bas, P., Furon, T., Cayre, F., Doërr, G., Mathon, B. "Watermarking Security", Springer, 2016.

[57] Quiring, E., Arp, D., & Rieck, K. (2018). Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking. In IEEE European Symposium on Security and Privacy.

[58] Song, C., Sudirman, S., Merabti, M., & Llewellyn-Jones, D. (2010, January). Analysis of digital image watermark attacks. In Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE (pp. 1-5). IEEE.

[59] Robin David, "LSB-Steganography: Python program to steganography files into images using the Least Significant Bit" [Online-2018] https://github.com/RobinDavid/LSB-Steganography

Authors' Profiles

Osama Hosam: Is a research associate in SRTA-City, Alexandria, Egypt. In 2007 he received his MSc. in computer systems and engineering from Azhar University, He pursued his PhD study in Hunan University, China and worked in parallel in Nanjing University of Technology; in 2011 he received his PhD in Computer Science and Engineering. In 2013 he worked as an Assistant Professor in at the Collage of Computer Science and Engineering in Yanbu. In 2017 he is promoted to be an Associate Professor in the field of Computer and Information Security, Taibah University. His research interests include, Computer Graphics, 3D

Watermarking, Stereo Vision, Pattern Recognition and Information security.

How to cite this paper: Osama Hosam, "Attacking Image Watermarking and Steganography - A Survey", International Journal of Information Technology and Computer Science(IJITCS), Vol.11, No.3, pp.23-37, 2019. DOI: 10.5815/ijitcs.2019.03.03