Borneo International Journal eISSN 2636-9826; Vol. 6 (4); 2023; 67-78
Published by *Majmuah Enterprise*
www.majmuah.com

Exploring Effective Attack Strategies on Steganographic Systems

ALAA JABBAR QASIM ALMALIKI[1], ROSHIDI DIN[1], SUNARIYA UTAMA[1]
, SALAM GHANIM NAJEEB[2], JABBAR QASIM ALMALIKI[3]
[1]*School of Computing, Universiti Utara Malaysia (UUM), 06010 Sintok, Kedah Darul Aman, MALAYSIA*
[2]*department of Medical Laboratories, College of Health and Medical Techniques, Sawa University, Almuthana, Iraq*
[3]*Ashur University, Department of Medical Instrumentation Technique Engineering College, Baghdad, Iraq*
Email :[1]alaa_jabbar@ahsgs.uum.edu.my,[1] roshidi@uum.edu.my, [1]sunariya.utama1@ahsgs.uum.edu.my ,
[2]Salam.alnajeb@yahoo.com , [3]Jabbar.qassim.f@au.edu.iq
| Tel : | [1]+60184020823 ,[1]+60175981306, [1]+601131548575, [2]+9647813000043, [3]+9647901530655|

Received: December 18, 2023
Accepted: December 21, 2023
Online Published: December 28, 2023

Abstract
First it is shown that the majority of steganographic applications for confidential communication have fundamental weaknesses. On the way to secure steganographic applications, the development of attacks to assess security is essential. The attacks on known algorithms presented here visualize the fallacy that the least significant bits are irrelevant. In addition, more objective methods for the detection of steganography using statistical means.

Keywords: assessment; summative; formative

1. Introduction

Steganography is an important argument in the debate about crypto regulation. The high level of confidentiality associated with encryption products such as e-mail, worries the security authorities, because in future state surveillance measures could lead to nothing. Any form of crypto regulation can easily be made more secure by using it steganographic procedures are circumvented - if they exist. In contrast to cryptography, in which a message is encrypted and then transmitted recognizable as an encrypted message the existence of the message cannot be proven with steganography and therefore cannot be prosecuted. In recent years, a variety of algorithms have been proposed with which messages in digitized images, video, audio and other multimedia data are translated. The majority of programs (e.g. S-Tools, Steganos, EzStego and Jsteg) hide messages by overwriting the least significant bits. As will be seen in the following, these algorithms have fundamental weaknesses. In the following sections, attacks with which the application of steganography is visualized will first be presented. This is followed by statistical attacks that are largely independent of the subjective impression of an observer. Steganography is also used for authentication (watermarking, fingerprinting). This opens up further important areas of application for steganographic methods, for the purpose of proving authorship . In the following, the term of steganography is used in the narrower sense and relates to confidential communication.

2. Steganography

2.1 Basics of Steganography

Steganography is the study of covert writing. Using steganographic methods it is possible to communicate confidentially. The advantage over cryptography (encryption of messages) is that beyond the confidential content, the existence of the message remains hidden. Figure 1 shows how a steady-state function works steganographic algorithm.

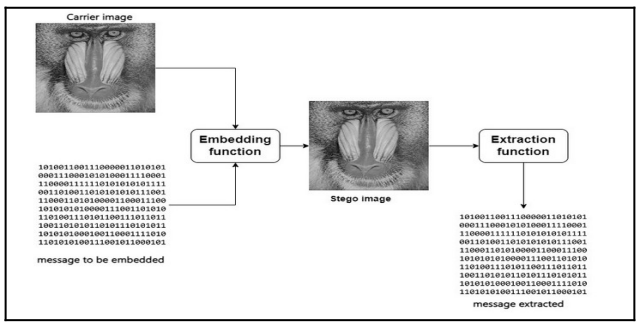

Figure 1: General functionality of a steganographic algorithm; the message to be embedded is hidden in a steganography and transmitted to the recipient, who can restore it

A steganography is generated on the transmitter side using the embedding function. The embedding function has two parameters: a carrier medium, which contains an indeterminacy (eg noise), and the message to be embedded. As a carrier medium, e.g. B. digitized images, video and audio data are used. The recipient can use the extraction function to restore the embedded message from the steganography. So that the use of a steganographic algorithm cannot be proven, the steganography must have statistical properties that cannot be distinguished from those of possible carrier media. Hence, both from the steganography and also read out a (potential) message from the carrier medium, because the steganographic algorithm cannot detect whether a message is received was embedded. From this it follows in turn that a message which is read from a steganography, not a potential one Statistically differentiates message from a carrier medium. There are some steganographic applications that are additionally powerful encrypt before embedding a message. This will make that "Secret" that keeps the message confidential, separated from the actual algorithm in the form of a parameter - the key. The steganographic algorithm can then be public. Only with the right key can it be decided whether the bits read from a potential steganography are really an encrypted message. A neutral uniform distribution is achieved through the encryption. In any case, it is advisable to encrypt the message to be embedded in a suitable manner, even with the other tools that do not do this implicitly. In order to decouple the security of the steganographic algorithms during the examinations from the form of the message to be embedded, only messages that are generated by a sufficiently strong pseudo-random generator are used in this work. Such messages represent the statistical properties of encrypted messages.

2.2. Algorithms Used

2.2.1 EzStego

GIF files contain a color palette that can contain up to 256 different colors from 2 Contains 24 possible, and the image content, a (LZW-compressed 3) Matrix of pallet indices. The message is encoded in the pixels without length information. The palette remains unchanged by the embedding. The steganographic algorithm reads the pallet and creates a copy. This copy of the palette is sorted. The colors must be sorted so that there are two adjacent colors in the sorted palette differentiate as little as possible. Sorting according to brightness is not always optimal , because two colors of the same brightness can also differ greatly in color. Every color can be understood as a point in a three-dimensional space. Figure 2 shows on the left the order of the colors in the RGB cube as they are stored in the palette of a GIF file, and on the right the order sorted by EzStego. 3 Lempel-Ziv-Welch

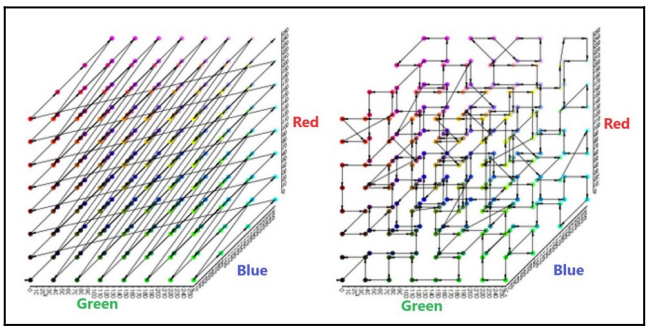

Figure 2: Color sequence in the palette (left) and sorted by EzStego (right); EzStego first looks for the shortest possible path through the RGB cube

In GIF files, colors are indicated by the three-color components r, g and b (for red, green and blue), each of which can assume values in the interval [0 ... 255].

Example

$$\left.\begin{matrix} r = 255 \\ g = 255 \\ b = 255 \end{matrix}\right\} \text{white} \qquad \left.\begin{matrix} r = 0 \\ g = 100 \\ b = 0 \end{matrix}\right\} \text{Dark green} \qquad \left.\begin{matrix} r = 165 \\ g = 42 \\ b = 42 \end{matrix}\right\} \text{Brown}$$

Borneo International Journal eISSN 2636-9826; Vol. 6 (4); 2023; 67-78
Published by *Majmuah Enterprise*
www.majmuah.com

The colors of a GIF file can assume a finite number of values in a limited three-dimensional space, the RGB cube. The smaller the spatial distance $a = \sqrt{\Delta r^2} + \sqrt{\Delta g^2} + \sqrt{\Delta b^2}$ between two colors in the RGB cube, the more difficult it is to distinguish them from one another, at least this is assumed by the sorting routine of EzStego as well as by most graphics processing programs. The sorting would be physiologically correct if the values of red, green or blue were adapted to the spectral light perception of the human eye. To do this, the values r, g and b must be weighted with the empirically determined coefficients 30%, 59% and 11%, respectively, determined by the International Commission on Illumination for additive color mixing, before the "spatial" distance in the colour Cube calculated will. On the sidelines of this work, a modified EzStego version developed with this weighting. With it, slightly less conspicuous results for the human eye could be achieved. The sorting routine determines, based on the first color in the original palette, the shortest path in the RGB cube over all colors down to as close as possible to the first color. The order of the colors on the shortest path is called the sorted palette. The pixels of a GIF file are palette indices that point to a color in the original palette: the color of the pixel. The colors of the original palette can be mapped bijectivity on the sorted palette see Figure 3. The pallet indices can also be applied to the sorted ones

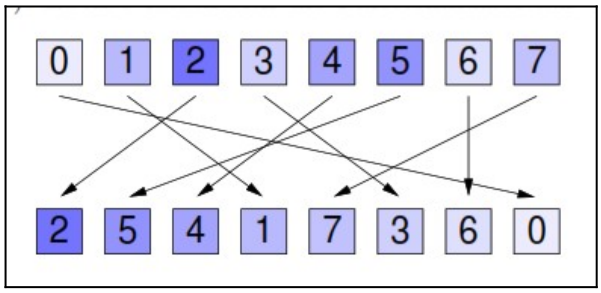

Figure 3: Mapping of the original pallet onto the sorted pallet

Map indices. A palette index and its corresponding sorted index point to the same color value, only in two different palettes. In general, therefore, they differ and can be converted into one another. The embedding function of EzStego processes the image content seamlessly, a matrix of pallet indices, starting at the top left and continuing downwards line by line. One bit is coded for each pixel. The first pallet index is converted into the corresponding sorted index. The least significant bit of the sorted index is replaced by the first message bit to be embedded. This new sorted index is converted into the corresponding index of the original pallet, the new pallet index. In the image content, the first pallet index is replaced by the new pallet index. The brightness and color of the pixel do not change at all or only very slightly see Figure 4. At this point it becomes clear why the colors are sorted at all: The change of the least significant bit of a pixel could cause a very drastic change in its color or its brightness if the original palette has an unfavourable color sequence. The embedding is done with the next pixel and the next message bit to be embedded continues until the last one has been processed.

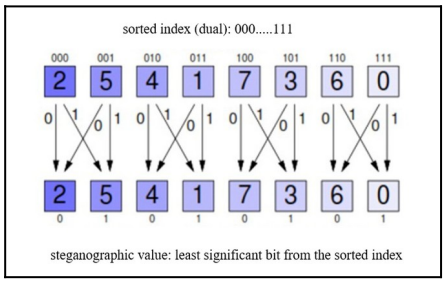

Figure 4: Embedding function of EzStego; assumed that the pallet index 7 is highlighted in the carrier image. Then it is replaced by the pallet index 3 when embedding a 1, and by 7 when embedding a 0.

2.2.2 Steganos

Steganos is an easy-to-use file hiding application. The Steganos algorithm comes from Fabian Hansmann. Images (BMP, DIB), sound files (WAV, VOC) and all kinds of ASCII texts or HTML documents are used as the carrier medium. The embedding algorithm first encrypts the data to be embedded with a modified RC4 stream structure. The bits of the encrypted stream are then used to overwrite the least significant bits of the carrier medium. The algorithm checks whether the The file fits into the medium, i. H. whether the capacity of the carrier medium is sufficient to hold the data to be embedded. If an attempt is made to embed more data than the capacity allows, Steganos aborts with an error message. If, on the other hand, less data is embedded, the stream charter continues to be spun until the capacity of

the image is fully utilized. Let us now turn to the true color BMP files as a carrier medium too. Steganos overwrites all least significant bits of the green components. The decision in Favor of green is not based on any investigations. Fabian Hansmann only adapted an older algorithm that only expects a gray value instead of the three-color components of BMP. The green color component is an unfavourable choice, since green is perceived as physiologically brighter than red and blue (see Section 2.2.1) and changes are perceived more strongly at the same intensity. True color BMP files are ideally suited as a carrier medium for steganography, the changes in color are visible to the human eye practically imperceptible. For EzStego, 256 colors are sufficient to produce barely perceptible differences. TrueColor BMP files can assign a color from 2 to 24 possible colors to each pixel. The subtlety of the color differences is thus beyond the limit that we can perceive. Figure 5 a carrier medium is compared to its steganography generated with Steganos .

Figure 5: The reunion as a carrier medium (l.) And Steganography (r.)

2.2.3 S-Tools

S-Tools can hide data in images and audio files. If the capacity of the carrier medium is not fully used, the steganographic algorithm spreads the message to be embedded over the entire length. The message bits are then password-controlled and pseudo-randomly distributed over the entire carrier file.

2.2.4 Jsteg

compresses images in the Jpeg-JFIF format as the carrier medium. Jpeg files contain lossy compressed image content. The brightness and color values of the pixels are initially in the frequency range transformed and then quantized (rounded). The transformation used is the discrete cosine transformation (DCT). This reduces the abundance of brightness values to a few DCT values that differ from zero. Jsteg replaces the least significant bits of the DCT values other than 0 and 1 with the message to be embedded.

3. Visual Attacks

This section describes the visual attacks developed by the author. Several authors of steganographic tools have independently assumed in the past that e.g. Least significant bits of luminance values in images can no longer be replaced. The mistake of this assumption is intended to be revealed by the visual attacks in this section. So far, messages have been embedded in digitized images using the majority of known steganographic algorithms by replacing carefully selected bits with bits of the message to be embedded will. The statistical properties of the least significant bits of digitized images were checked by Elke Franz and could not be distinguished from random bits by statistical means. It is actually difficult to distinguish image content from noise by statistical means. And it is even more difficult to distinguish the least significant bits of a digitized image from random bits. The difficulty lies in defining permissible image content as a formal quantity. It is intuitively clear to sighted people what the content of the picture is. However, the boundary is blurred and also depends on the imagination - who has not already discovered shapes in a cloud formation? The human eye is practically trained to recognize known things. This human ability is a prerequisite for the visual attacks developed here.

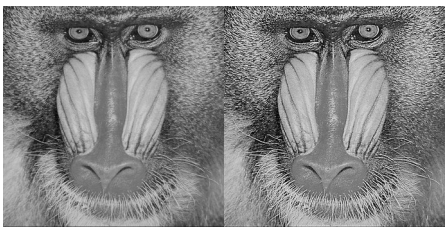

Figure 6: Windmill as a carrier medium (left) and steganography (right); Steganography hardly noticeably changes the carrier medium.

In Figure 6 only a very slight difference can be seen between the carrier medium and the steganography. There is nothing in the left picture embedded. A 3600 byte long, random message with EzStego was embedded in the picture on

the right. Without a direct comparison of the two images, the changes would probably not arouse suspicion. In general, only the changed carrier file is transmitted, so that a direct comparison is impossible.

3.1 Steganographic Value of a Pixel

Each bit provided for change by a steganographic algorithm has a location within the carrier medium - this refers to the location visible in the image, which can also be specified in pixel coordinates, not the position within the file. Each of these changeable bits is therefore assigned to a pixel. Several changeable bits can be assigned to one pixel. Let us now consider an image that has been modified with a carefully developed steganographic tool, we cannot see any noticeable changes. Often it is not even possible for us to apparently distinguish the original and the altered image. It is, however, possible to read out the message supposedly contained in the changeable bits and to display it graphically. If the steganographic algorithm embeds a bit for each pixel, we can display the steganographic value of the bit contained in the pixel, e.g. white for the value 1 and black for the value 0. There are also Algorithms that embed more than one bit per pixel can then we represent the steganographic value of a pixel with additional colors.

3.2 Idea of The Visual Attack

The idea of this attack was born by accident. The author wanted to find out whether a contact address might be hidden on the homepage of “a pseudonymous hacker on the Internet”. 5 First, the single image on the page was examined for steganographic content by reading out the potential message with various tools. For example, EzStego delivered a supposed message that was not readable ASCII text. The search for the contact address was unsuccessful in the picture. In the hexadecimal representation of the supposed message bytes, it was noticeable that characters were often repeated. A comparison with the carrier medium showed that the repetitions occurred exactly where there are uniform surfaces in the image. Let's try to understand the situation using the example. If we extract the supposed message with EzStego from the picture on the left in Fig. 6 - although we know that no message has been embedded - we do not get any readable text, but a series of bytes that can be represented in hexadecimal. In Fig. 7 the hexadecimal representation has been formatted so that there are 240 bits on each line (because the carrier medium is 240 pixels wide). The complete representation would be longer

```
ffffffffffffffffffff40ff7fffffffffffffffffffffffffffffff
ffffffffffffffffffffe0179fffffffffffffffffffffffffffffff
ffffffffffffffffffff8dcfbfffffffffffffffffffffffffffffff
ffffffffffffffffffff21fffffffffffffffffff77fffffffffffff
ffffffffffffffffffffc04fffffffffffffffffefbfffffffffffff
ffffffffffffffffffffe03fffffffffffffffffebfffffffffffff
ffffffffffffffffffffff087dfffffffffffffffba1fffffffffffff
ffffffffffffffffffff8827fffffffffffffffe754fffffffffffff
ffffffffffffffffffffff9a1fffffffffffffffdf77fffffffffffff
ffffffffffffffffffffffc04fe3fffffffffffff7ea5bfffffffffff
ffffffffffffffffffffffc2058fffffffffffffeba87bfffffffffff
ffffffffffffffffffffffe28e77fffffffffffdf91f5fffffffff...
```

Figure 7: Hexadecimal representation of a supposed message; the alleged message read out of the steganographic ally unchanged carrier medium shows "WindmÃ¼hlen".

as one print page (240 lines). Nevertheless (with a little imagination) the tips of two windmill blades can be seen in the first lines. Sequences of 00 (eight 0-bits) and ff (eight 1-bits) occur particularly frequently, which is characteristic of digitized black-and-white images. What could be more natural than trying to depict the alleged message in pictures.

3.3 Visual Embedding Filter

3.3.1 EzStego

An embedding filter depicts the steganographic values of the pixels described in Section 3.1. With EzStego, the colors of the pixels (and thus the steganographic values) are determined by the palette. The embedding filter for EzStego replaces the original palette with a black and white palette. Figure 8 shows how the colors are replaced. Depending on the sorted index, the colors are replaced by black or white. The palettes of the images in Figure 9 have been replaced. It can now be seen without difficulty that the upper half of the right picture is noisy, while in the lower half, as in the left picture, the windmill can be seen like a shadow. That

Borneo International Journal eISSN 2636-9826; Vol. 6 (4); 2023; 67-78
Published by *Majmuah Enterprise*
www.majmuah.com

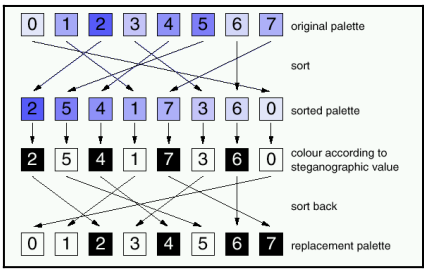

Figure 8: Assignment function of the substitute colors; Colors that have an even index in the assorted palette are replaced by black, those that have an odd index are replaced by white.

The result of the visual attack shows that a message was most likely embedded in the image on the right. The weaknesses of the steganographic tool EzStego can be identified:

1. The bits of the carrier medium that are replaced by message bits are statistically dependent on the carrier medium - our eyes can clearly see the connection with the original image.
2. If the embedded message is shorter than the maximum possible Message length, only the upper part of the picture is used for embedding the used. This means that in the filtered image there is a horizontally running border recognizable. The limit allows an estimate of the Length of the embedded message. In our example the Border in the middle between the upper and lower edge of the picture. That The image of the windmill has a format of 240 × 240 pixels. In EzStego can hide a bit for each pixel (see section 2.2.1). So about 240 x 120 = 28,800 bits or 3,600 bytes are hidden. Below this limit, image content may still be recognizable; above this, the filtered image is completely independent of the original (provided - and this is generally the case - the embedded message is independent of the carrier medium).

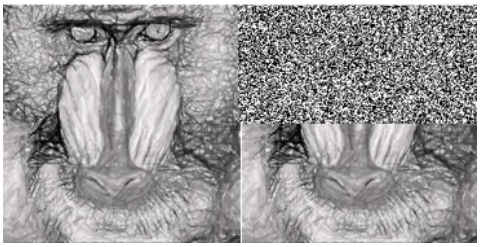

Figure 9: Embedding filter applied to Fig. 6; an embedded message (right) is stochastically independent of the carrier medium.

3.3.2 Steganos

In Figure. 5, only one byte was embedded in the steganography. That is far less than the capacity of the picture allows. Both images in Figure. 5 have the format 356 × 239. The capacity of the image is one bit per pixel about 10 KB. Shorter messages are lengthened by the continuation of the flow chart in such a way that all pixels are used. This becomes clear in Figure. 10. 6 With EzStego, an embedded byte would only affect the first eight pixels. The embedding filter for Steganos only shows pictures that are completely noisy. The comparison residue that occurred with EzStego and the jump between noise and image content at the end of the message to be embedded are missing.

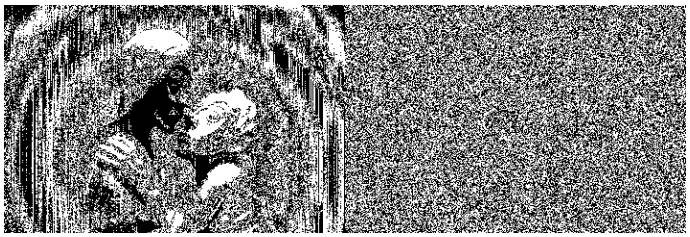

Figure 10: Embedding filter applied to Figure. 5; Steganos leaves its traces in all pixels (right).

The steganographic embedding algorithm for the true color BMP format is incorrectly implemented in Steganos. He disregards the peculiarity that the pixel lines in BMP files are always a multiple chess is 4 bytes long. To do this, the image lines are padded with zeros until a 32-bit word limit is reached. Figure. 11 shows four different ones

5 pixels	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	0	1 Fill change

4 pixels	R	G	B	R	G	B	R	G	B	R	G	B	0 Fill change
3 pixels	R	G	B	R	G	B	R	G	B	0	0	0	3 Fill change
2 pixels	R	G	B	R	G	B	0	0					2 Fill change

Figure 11: Lines of pixels of various lengths in true color BMP files; In true color BMP files, the bytes of each pixel line are padded to a 32-bit word limit.

Length of picture lines with the necessary full bytes. Steganos assumes that the first pixel of the following line immediately follows the last pixel of a line. The result is that Steganos also embeds it in the full bytes. If a picture has three full bytes per picture line, then these are interpreted by Steganos as a color component for red, green and blue. The first value of the following picture line is correctly interpreted as red. This changes if one or two full bytes are appended: Steganos then embeds color values that change from line to line. On average, every sixth full byte is converted from 0 to 1. Whenever a 1 appears as a full byte, this is a suspicion of steganography arousing circumstance.

Figure 12: Carrier medium and steganography with format-related full bytes; the carrier medium now has the format 358 × 239 and internally contains 478 full bytes. Steganos sets around 80 of them to 1 - invisible, but verifiable.

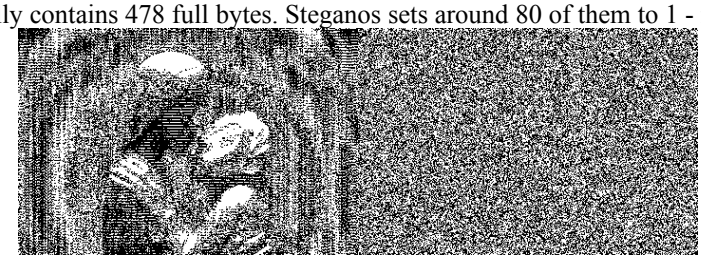

Figure 13: Embedding filter applied to Figure. 12; Steganos changes the colour component used from line to line. Horizontal stripes can therefore be seen in the filtered carrier medium (left).

3.3.3 S-Tools

In order to be able to embed 256 colors in BMP files, the S-Tools redesigned the color palette very conspicuously. Three bits of the message are embedded per pixel. To do this, the number of colors in the palette is reduced. Fig. 14 shows two images on VIS'97 as an example

Figure 14: Carrier medium (left) and S-Tools steganography (right); the S-Tools embed three bits per pixel. The 256 colors of the palette are reduced to 32 colors.

were used to build the Windows NT component of the basic protection manual, also. 120 pages of text with images can be embedded in a 700 KB BMP file without being noticed. In direct comparison with the original, it is noticeable that the color palette has been reduced. Instead of 256 different colors, the palette in the steganography contains 32 × 8 colors, with 8 colors only differing from red, green and blue by the three least significant bits. These 8 colors cannot be distinguished with the naked eye. Such palettes are so conspicuous that an examination of the image content is not

necessary. Nevertheless, a visual attack is possible and very impressive, since three message bits per pixel produce a very clear, filtered image. The filtered image (Figure. 15) contains eight different colors.

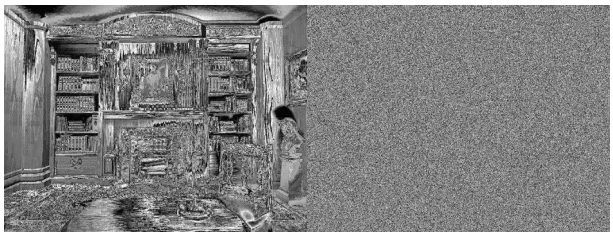

Figure 15: Embedding filter applied to Figure. 14; three bits per pixel can provide eight steganographic values. This creates a particularly clear, filtered image.

3.4 Limits to Visual Attacks

Since the visual attacks are based on the recognizability of the image content, there are limits to them. Figure. 16 shows e.g. a floor panel whose

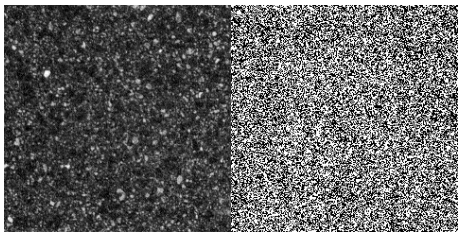

Figure 16: Floor plate, original and filtered; Nothing was embedded in this image, although the filtered image is noisy. Visual attacks are only meaningful if the image content is sufficient.

Grain is very random and even. Although nothing is embedded in the carrier medium shown, the filtered image looks noisy.

4. Fundamentals of Mathematics

The limits of visual attacks, which became clear in Section 3.4, can be overcome with statistical means. In order to understand the statistical attacks developed for this purpose, the mathematical tool is introduced in this section, which can be found in the current literature on the fundamentals of statistical analysis.

4.1 Chi-Square Distribution

In connection with Gauss's theory of errors, the Astronomer Helmert sums of squares of quantities that are normally distributed. Pearson later called the distribution function proven chi-square distribution (χ^2 -distribution). From the stochastically independent, normalized normally distributed random variables X_1, X_2, ..., X_f we form the sum of squares

$$X_f^2 = X_1^2 + X_2^2 + X_f^2 \quad for\ f = 1,2, \ldots.$$

The random variable x is continuous and has density

$$d_f(x) = \begin{cases} 0 & \text{For } x \leq 0, \\ \frac{1}{\frac{f}{2^2}\Gamma\left(\frac{f}{2}\right)} e^{-\frac{x}{2}} x^{\frac{f}{2}-1} & \text{For } x > 0, \end{cases}$$

Here $\tau(a) = \int_0^\infty e^{-t} t^{a-1}$ is the so-called gamma function. Partial integration provides the relationship

Borneo International Journal eISSN 2636-9826; Vol. 6 (4); 2023; 67-78
Published by *Majmuah Enterprise*
www.majmuah.com

$$\tau(a+1) = a\tau(a)$$

For $a = \frac{1}{2}$ and a = 1 applies specifically

$$\tau\left(\frac{1}{2}\right) = \sqrt{\pi}\,;\ \tau(1) = 1$$

From (2) and (3) it follows for every natural number n
$$\tau(n) = (n-1)$$

The distribution of the random variables χ 2f is called the chi-square distribution with f degrees of freedom. Equation (1) is shown in further literature with the help of some conversions by complete inductio. The calculation of the quantiles of the chi-square distribution is necessary for the chi-square test. The chi-square distribution was implemented in the Java programming language as the basis for the statistical tests. The quantiles are determined by numerical integration according to Simpson.

4.2 Chi-Square Test

To compare distributions, as will be done in the next section 5, we use what is probably the most well-known and important test of goodness, the so-called chi-square test. It is based on a comparison of the empirical frequency distribution F(x) obtained from a random sample $x_1, x_2, ..., x_n$ with the theoretically expected distribution $F_0(x)$ of the population from which the sample originates. We test the null hypothesis

$$H_0: F(x) = F_0(x)$$

against the alternative hypothesis

$$H_0: F(x) \neq F_0(x)$$

To do this, we proceed step-by-step:

1. Subdivision of the sample into classes and determination of the absolute Second-class frequencies (occupation numbers): Experience has shown that each class should contain at least 5 sample values.
2. Calculation of the theoretically expected absolute class frequencies
3. Determination of a suitable measure for the deviation between the observed and the theoretical distribution: Let ni be the empirical absolute frequency in the i-th class and n ∗ i

the theoretically expected absolute class frequency. According to Pearson, a suitable measure for the deviation between the observed and the theoretical distribution is the measure

$$X^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - n_i^*)^2}{n_i^*} = \sum_{i=1}^{k} \frac{(\Delta n_i)^2}{n_i^*}$$

which satisfies the chi-square distribution with f = k m 1 degrees of freedom, if m is the number of unknown parameters estimated with the help of the sample. 4. Calculation of the test value χ2 k 1 and determination of the corresponding p-value.

5. Statistical Attacks

5.1 EzStego

The embedding function of EzStego overwrites the least significant bits of the sorted indices. If the least significant bits are overwritten, then two values - we shall call these pairs - are merged into one another, which differ only in the least significant bit. If the bits that replace the least significant bits are evenly distributed, then the frequencies of the values of a pair are equalized. Figure 17 goes back to the example from Figure. 4 and shows how the frequencies of colors in a picture change when a uniformly distributed message is embedded with EzStego will. The columns with an even sorted index are called

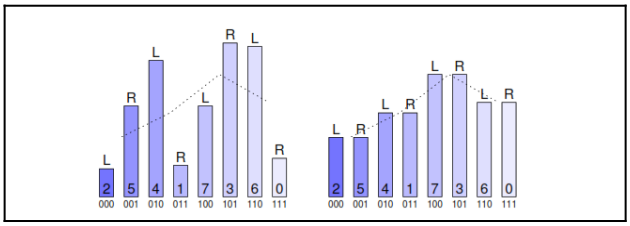

Figure 17: Color histogram before and after embedding with EzStego; By embedding with EzStego, color pairs are created, the frequency of which is compensated.

Borneo International Journal eISSN 2636-9826; Vol. 6 (4); 2023; 67-78
Published by *Majmuah Enterprise*
www.majmuah.com

Left distributions, those with odd, are called right distributions (marked with L and R in Figure. 17). Before embedding with EzStego, the two distributions are different, after embedding they are aligned with one another. The idea of the statistical attack is that one Compare the left distribution with the distribution expected after embedding. Normally, the expected distribution should not be determined from the sample. However, it agrees exactly with that from the original image, from which it actually had to be determined. The expected frequency depends on the sum of the two frequencies of a pair. This sum remains unchanged by embedding with EzStego (although the summands change). The original image is therefore not necessary for this attack. The degree of agreement is a measure of the embedding probability. The agreement is with the Chi-square test determined in the following steps (see section 4.2):

1. The k classes are all palette indices whose referenced color cx, y is stored in the sorted palette at an even index. Their occupation number $n_i = | [C_{x,y} | \text{sortedIndexOf}(C_{x,y}) == 2_i] |$ In the case of embedding, a theoretically expected class frequency $n * I >$ Own 4.
2. The theoretically expected class frequency after embedding a uniformly distributed message result from
$$n_i^* = \frac{|C_{x,y}|\text{sortedIndexOf}\left(C_{x,y}\right) \in \{2i, 2i+1\}\}|}{2}$$
3. The measure of the deviation is $X_{k-1}^2 = \sum_{i=1}^{k} \frac{(n_i - n_i^*)^2}{n_i^*}$
4. The p-value is determined by integrating the density function: $p = 1 - \int_0^{X_{k-1}^2} d_{k-1}(x)dx$

Figure. 18 shows on the left a steganography in which a 3,600 bytes long message has been embedded, the same message as in Figure. 6 on the right Attack (filtered picture on the right) to its limits. The slide

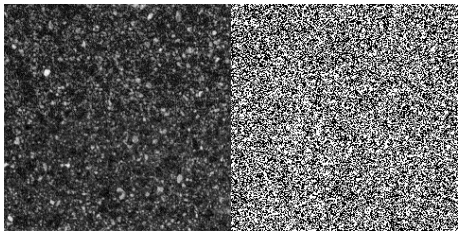

Figure 18: Floor slab as EzStego steganography and filtered; a visual attack cannot distinguish the upper, steganographic half from the lower, original half.

The graph in Figure. 19 shows the embedding probability (the p-value) as a function of an increasingly extensive sample. The sample initially comprises 1% of the pixels, starting from the upper edge of the image. For this sample, the embedding probability calculated using equation (4) is p = 88.26%. The next random sample also includes a further percent of the image points, a total of 2% of the total image, and the p-value increases to 98.08%. As long as the sample only contains pixels in the upper half of the image in which a message was embedded, the graph does not drop below 77%. The pixels in the lower half of the image are unchanged because the message to be embedded is not was longer. A sample of 52% of the pixels contains enough unchanged pixels to allow the graph to sink to an embedding probability of practically 0 (" practical" means in this context that the remaining probability is smaller than the calculation accuracy is).

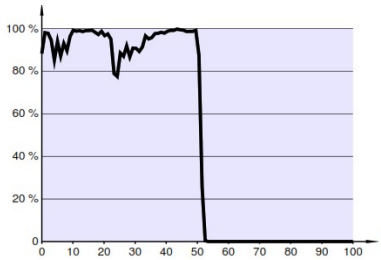

Figure 19: Embedding probability for EzStego Figure. 18; a statistical analysis of the steganography from Figure 18 shows steganographic changes.

5.2 Steganos

Steganos uses I.A. True color images in BMP format as a carrier medium and provide a capacity of one embedded bit per pixel. Regardless of the length of the data to be embedded, the least significant bits of all green components are overwritten with bits that are stochastically independent of the image content. So, on average, every second pixel is

changed. Even if only one byte is embedded - that is the minimum length for the message to be embedded that Steganos allows - the message is obviously extended until it contains as many bits as there are pixels in the carrier medium available. The embedding function of Steganos replaces the least significant bits of the green parts of each pixel with a message bit (or a padding bit). The green component is represented by 8 bits per pixel. A carrier medium therefore contains a maximum of 256 different values in the green component. Since all the green parts in the carrier medium are processed during embedding, the population from which we obtain a sample comprises the entire image.

1. The k classes are all even values in the green part $g_{x,y}$, whose occupation number $n_i = |[g_{x,y} \mid g_{x,y} = 2_i]|$ if one is embedded theoretically has expected class frequency $n * i > 4$.
2. Steganos converts the message to be embedded into evenly distributed bits. Therefore, when embedding these bits, as if the least significant bits of the green components are overwritten with these evenly distributed bits, the theoretically expected class frequency is adapted to the following value:

$$n_i^* = \frac{\left|g_{x,y}|g_{x,y}=2_i\right| + \left|\{g_{x,y}|g_{x,y}=2_i+1\}\right|}{2}$$

3. The further steps for calculating the p-value agree with those in Section 5.1.

In practical studies, the p-value rarely deviated from 0 or 100% away.

5.3 Jsteg

The steganographic changes are made in the frequency range. If a least significant bit of a DCT value is changed, this affects up to 16×16 pixels. The changes are superimposed in several pixels, so no visual attack is possible. The embedding function of Jsteg overwrites the least significant bits of the DCT values other than 0 and 1. The frequencies of amplitude values that differ only in the least significant bit are balanced. In contrast to EzSteganos and Steganos, the following also applies that (if nothing is embedded) the smaller value is expected more often than the partner value differing in the least significant bit.

1. The k classes are all even amplitude values a that differ from 0. Its occupation number $n_{(i+2048)} = |\{a \mid a = 2i \wedge i = 0\}|$ must with embedding a theoretically expected class frequency $n * i > 4$ own.

2. The theoretically expected class frequency after embedding results through

$$n_{(i+2048)}^* = \frac{|\{a|a\in\{2_i,2_i+1|i\neq 0\}\}|}{2}$$

3. The further steps for calculating the p-value agree with those in Section 5.1.

Practical investigations showed that the p-value was just as highly selective as with Steganos.

6. Conclusions

The strategy for embedding most steganographic systems of overwriting the least significant bits is at best resistant to apparent attacks. On the one hand, the visual attacks showed that the least significant bits are generally not irrelevant, as is required by today's steganographic applications. On the other hand, it has been shown that irrelevance alone is not enough to prevent attacks. Overwriting the least significant bits leaves traces as statistically independent frequencies are compensated. Statistical attacks are superior to visual attacks because they are not based on the image content of the carrier file. As the embedding rate decreases (in the simplest case by spreading the data to be embedded), the probability of error of the attack increases. Unfortunately, the relative throughput drops and the algorithm works less electively. In the extreme case, we have an application that is completely secure but does not embed anything. Analogous to the development of cryptography, the presented took a step in the iterative process that leads to more secure steganographic systems. Better algorithms should use the operation replace writing with others, e.g. B. by incrementing. Frequencies are then not balanced out, but rather circulate in the range of values. The steganographic changes must also remain limited to the noise maxima of the carrier medium and should not be localized there deterministically. As an alternative to the iterative process (attack defines), non-deterministic phenomena from input devices can be used, such as that of a camera or a scanner. The examination of input devices shows free spaces that allow the embedding of data. If stenographic techniques simulate the peculiarities of a camera, this will be the case do not raise doubts to a potential attacker.

References

Alyousuf, F. Q. A., Din, R., Qasim, A. J. J. B. o. E. E., & Informatics. (2020). Analysis review on spatial and transform domain technique in digital steganography. 9(2), 573-581.

Anderson, R. (1996). Information Hiding: First International Workshop Cambridge, UK, May 30–June 1, 1996 Proceedings. Paper presented at the International Workshop on Information Hiding 1.

Bosch, K., Bosch, K., Bosch, K., Bosch, K., & Mathematician, G. (2006). Elementare Einführung in die Wahrscheinlichkeitsrechnung (Vol. 6): Springer.

Din, R., Bakar, R., Utama, S., Jasmis, J., Elias, S. J. J. B. o. E. E., & Informatics. (2019). The evaluation performance of letter-based technique on text steganography system. 8(1), 291-297.

Din, R., Ghazali, O., & Qasim, A. J. J. I. J. E. E. C. S. (2017). Analytical review on graphical formats used in image steganographic compression. 5(3), 401-408.

Din, R., Ghazali, O., Qasim, A. J. J. I. J. o. E. E., & Science, C. (2018). Analytical review on graphical formats used in image steganographic compression. 12(2), 441-446.

Din, R., Mahmuddin, M., Qasim, A. J. J. I. J. o. E., & Technology. (2019). Review on steganography methods in multi-media domain. 8(1.7), 288-292.

Din, R., Qasim, A. J., Abdullah, S., Elias, S. J. J. I. J. o. E., & Technology. (2019). Analysis Review on Image Compression Domain. 8(1.7), 293-296.

Din, R., Qasim, A. J. J. B. o. E. E., & Informatics. (2019). Steganography analysis techniques applied to audio and image files. 8(4), 1297-1302.

Din, R., & Utama, S. J. B. I. J. e.-. (2023). The Design Review of Feature-based Method in Embedding the Hidden Message in Text as the Implementation of Steganography. 6(3), 88-95.

Din, R. J. B. I. J. e.-. (2023a). Comparative Analysis of Methods for Digital Steganography in Images. 6(3), 119-127.

Din, R. J. B. I. J. e.-. (2023b). Comparison Of Steganographic Techniques of Spatial Domain and Frequency Domain in Digital Images. 6(3), 109-118.

Franz, F. E. J. G. B., TU Dresden, Institut für Theoretische Informatik. (1996). Untersuchung von Bewertungsmöglichkeiten für Bilder bezüglich eingebettteter steganographischer Daten.

Gellert, W., Kästner, H., & Neuber, S. J. (1981). Lexikon der mathematik.

Kuhn, M. G., & Anderson, R. J. (1998). Soft tempest: Hidden data transmission using electromagnetic emanations. Paper presented at the Information Hiding: Second International Workshop, IH'98 Portland, Oregon, USA, April 14–17, 1998 Proceedings 2.

Nelson, M. R. J. D. D. s. J. (1989). LZW data compression. 14(10), 29-36.

Qasim, A. J., & Alyousuf, F. Q. A. J. Q. Z. J. (2021). History of image digital formats using in information technology. 6(2), 1098-1112.

Qasim, A. J., Din, R., Alyousuf, F. Q. A. J. B. o. E. E., & Informatics. (2020). Review on techniques and file formats of image compression. 9(2), 602-610.

Razali, N., Mustapha, A., Utama, S., & Din, R. (2018). A review on football match outcome prediction using bayesian networks. Paper presented at the Journal of Physics: Conference Series.

Reinhardt, F., Soeder, H., & Falk, G. (1992). dtv-Atlas zur Mathematik, Band 2: Analysis und angewandte Mathematik. In: Deutscher Taschenbuch Verlag GmbH & Co. KG, München.

Renyi, A. (1962). Wahrscheinlichkeitsrechnung VEB Deutscher Verlag der Wissenschaften. In: Berlin.

Utama, S., Din, R. J. J. o. A. R. i. A. S., & Technology, E. (2022). Performance Review of Feature-Based Method in Implementation Text Steganography Approach. 28(2), 325-333.

Welch, T. A. J. C. (1984). A technique for high-performance data compression. 17(06), 8-19.

Westfeld, A., & Pfitzmann, A. (1999). Attacks on steganographic systems: Breaking the steganographic utilities EzStego, Jsteg, Steganos, and S-Tools-and some lessons learned. Paper presented at the International workshop on information hiding.

Westfeld, A., & Wolf, G. (1998). Steganography in a video conferencing system. Paper presented at the International Workshop on Information Hiding.

Ziv, J., & Lempel, A. J. I. T. o. i. t. (1977). A universal algorithm for sequential data compression. 23(3), 337-343.