IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SECURED DATA TRANSMISSION ON VIDEO STENOGRAPHY

[1]V D Venkatraman, [2]K Mohana Krishna

[1]PG Student, [2]Assistant Professor

[1]VLSI Design,

[1]Shree Institute of Technical Education, Tirupati, India

Abstract: It is very essential to transmit important data like banking and military information in a secure manner. Video Steganography is the process of hiding some secret information inside a video. The addition of this information to the video is not recognizable by the human eye as the change of a pixel color is negligible. This paper aims to provide an efficient and a secure method for video Steganography. The proposed method creates an index for the secret information and the index is placed in a frame of the video itself. With the help of this index, the frames containing the secret information are located.

Hence, during the extraction process, instead of analyzing the entire video, the frames containing the secret data are analyzed with the help of index at the receiving end. When steganographed by this method, the probability of finding the hidden information by an attacker is lesser when compared to the normal method of hiding information frame-by-frame in a sequential manner. It also reduces the computational time taken for the extraction process.

Index Terms **– Data transmission, security, video processing, stegnography**

I. INTRODUCTION

The word Steganographic means “covered writing” and comes from the Greek words steganos, which means covered or secret and graphy, which means writing or drawing. Steganographic is the “cousin” of cryptography and is the art of covert communications. Its roots go all the way back to the ancient Greeks when the fellow named Histiaeus, a prisoner of a rival king, needed to pass a secret message to his army. To accomplish that he shaved the head of the trusted slave, tattooed the message on it and sent him to deliver the message in person when his hair grew back (Cole). Other early forms of Steganographic include writing on wood under the wax on the wax tablet, using invisible ink made out of juice or milk which shows when heated and using the method called “microdots” which reduces a text to a size of a single dot and inserts in a document as punctuation. Today, Steganographic is usually referred to as hiding data in digital media(Cole).

For example, a text document can be hidden in an image, sound or another text file. This paper concentrates specifically on data hiding in images. First, it presents some terminology and background information about images and some of their formats. Next, different methods for Steganographic will be discussed which are publicly available and widely used today. Following methods, specific tools that implement those methods will be described along with the results of tests performed on them to verify their quality. Concluding section will sum up what was discussed and give a look ahead to where the field is headed.

There are some terms that will be used throughout the paper and here they are defined for easier understanding of the subject. The image in which the data is going to be hidden is called the cover image(Johnson, Jajodia), also called carrier image, overt or host file(Cole). After the data, which is referred to as covert data(Cole), has been hidden in an image, that image is called stego-image(Johnson, Jajodia), also sometimes called stego-carrier or stego-medium(Johnson, Jajodia). Stego-key(Johnson, Jajodia) can be used to make the stego-image more secure; it is the password that is stored in the carrier file along with the covert file without which data cannot be extracted from the stego-image. Looking for possible information in images and decoding it is called steganalysis.

To understand how it is possible to hide files in other images it is important to know what digital images are comprised of. As other types of digital media, digital images are nothing but an array of numbers, representing light intensities at various picture elements or pixels(Cole). Each pixel is represented by some number of bits, now usually 8 or 24. With 8-bits, there is a total of 256 colors available. Every image using this representation has a color palette saved with the image which contains every color available to the image(Kessler). The common digital image format using 8-bit colors is Graphics Interchange Format (GIF). With 24-bits, color is derived from the percentage combination of the three primary colors – red, green and blue, 8 bits for each for a total of 265*256*256 different colors. They can be represented as hex, decimal or binary.

For example, the color white can be represented as such: 11111111, 11111111, 11111111 – 100% for red, 100% for blue and 100% for green.(Johnson, Jajodia) The common formats that use it are BMP(Bitmap) and TIFF(Tagged Image File Format). There also are grayscale images that use 256 shades of gray and pure black and white which use only 2 colors where each pixel can be represented with 1-bit. The size of the image file depends on the number of bits used to represent the colors and the number of pixels in the image. The higher the both numbers, the higher the resolution of the image and the more space is needed to store such a file. For example, a 24-bit image of 1,024 by 768 pixels requires more than 2 MB of space (Johnson, Japonica).

This is still a lot to transmit over the internet, thus different compression methods have been invented. There are two categories of compression: lossless and lossy. Lossless formats use mathematical algorithms to manipulate data and compress images such that the original image can be recovered exactly when decompressing. It reduces the storage space needed by about 50%, which might still not be enough. Its mostly used to compress text and other data files where every piece of information is important and image file formats GIF and BMP. Lossy compression on the other hand can reduce storage required by 90% without noticeable to human eye changes and down by 99% with noticeable changes when size of the file matters more than quality. The most popular format that uses it is JPEG (Joint Photographic Experts Group¬), which cuts out the non-significant data from images. The original file cannot be recovered exactly when using lossy compression, that's why it is best to use lossless when embedding files into images so that if the stego-image is recompressed the embedded data will not be lost.

Next, there are many different methods for hiding information in digital files. They are categorized in three groups: insertion-based, substitution-based and generation-based. Insertion-based techniques hide information in fields of the file which is not used by the program. It is used mostly to hide data in a text document with EOF (end-of-file) markers. Most text editors ignore any information after that marker, so anything of any size can be inserted there, however, those insertions do change the file size and if the editor only displays 2 lines of text and the size of the file is big, then it might raise suspicion. Another technique is substitution-based and involves substituting covert data in place of insignificant data in the overt file.

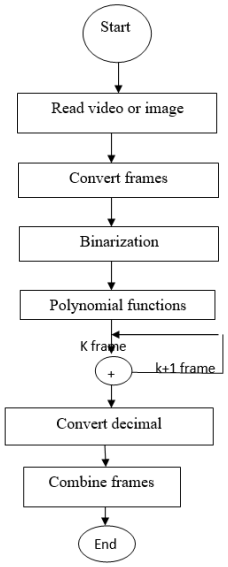
The resulting stego-carrier is the same size as the overt file, but the space for putting covert data is limited. Last one is generation-based. This technique does not need the overt file, it only needs the secret file. The covert file is used to determine the angle, length, and color of each line that will comprise the newly generated image by using mathematical-based manipulation. Such image will most likely not be of a known object (for example a house or a person) but rather a random-looking image with a collection of groupings of similar colors. It will not necessarily raise suspicion since people might think it's just a work of abstract art or something else not known by them. Also, even with suspicion, it would be fairly difficult to detect that it is a stego-image with detection tools available today.(Cole) Out of these three techniques, substitution-based is used the most today thus it is discussed further in the next section.

The most popular and widespread method of substitution-based technique is the LSB (Least Significant Bit) substitution. It works better on the 24-bit images because of the greater color choices. It changes the 1st and sometimes 2nd least significant bits (bits that have no noticeable effect on the image) without disrupting the appearance of the image, because it will change the pixel only by a shade of the color. With 8-bit images, color choices are limited, and bits actually serve as pointers to the color palette, so changing them would make pixels reference a different color. It is still possible to hide secret data in them, but the cover-image has to be selected more carefully(the tips for selecting cover images are presented later).(Johnson, Jajodia) This method is used on lossless compression formats due to the fact that with lossy compression least significant bits are likely to be discarded. Another approach is masking. It taxes pixel values in masked areas and raises or lowers them by some percentage, which makes the mark invisible. Similar to it is patchwork method, in which pairs of patches are selected pseudo-randomly and pixel values in one patch are raised and lowered in another by the same value. (Wang, Wang).

In this paper, two novel methods for selecting the bytes of the cover medium, in which the secret data to be stored are proposed. The secret data is stuffed in the least significant bits of the selected bytes. The first method is based on a polynomial function of order one, which is used to determine the byte of the cover medium where the secret data is going to be stored. The second method is based on a Quadratic polynomial function which determines the bytes of the cover medium to be used for storing the secret message. And the paper discusses the advantages of above discussed methods.

Currently most of the steganography algorithms work by modifying the Least Significant Bit (LSB) of the consecutive bytes of the cover medium to store the secret data [2]. The main drawback of this algorithm is that hidden message can be retrieved easily through steganalysis since the messages are stored in consecutive bytes. Steganalysis is the method by which to detect the presence of a hidden message and attempt to reveal the true contents of the message [3] .Steganographic technique embed a message inside a cover. Various features characterize the strength and weaknesses of the methods. The relative importance of each feature depends on the application. They are Capacity, Robustness, Invisibility, and Security.

II. IMPLEMENTATION



Reading of frame from a video sequence

Firstly I taken a video and then read all the frame in the video by the matlab function Aviread.The process of this is explained below.

a) Access Video with MMREADER
The mmreader function constructs a multimedia reader object that can read video data from Multimedia file. I used the following function to read all my videos. This function is very Useful as it can read the video file with different formats which are not been read by simple Function like AVIREAD. mmreader supports the following formats: AVI, MPG, MPEG, WMV, ASF, and ASX
videoFile = mmreader(PATH);

b) Reading of the frame from the object created by mmreader
After the object created by mmreader .The object is used for reading the video file so that frame can be extracted from it and the processing can be done on each of the frame. The function for the same is described below-:

Movie= aviread(videoFile,Frameno);

The function aviread is used to extract a frame from the video, where movieFile is the path of the movie file and frameNo is the number of frame to be extracted. The function will extract the entire frame and will store it into the Movie.

c) Getting number of frames
To get number of frame from a video I use the following function.

numFrames = get(videoFile, 'numberOfFrames');

The above function will extract the number of frame in the video file and will store it into the variable name numberofFrames.
d) Frame conversion
To convert frames from video, follow given procedure
readerobj = mmreader('xylophone.mpg', 'tag', 'myreader1');
Read in all the video frames.
vidFrames = read(readerobj);
Find out how many frames there are.
numFrames = get(readerobj, 'numberOfFrames');
Create a MATLAB movie structure from the video frames.
for k = 1 : numFrames
mov(k).cdata = vidFrames(:,:,:,k);
mov(k).colormap = [];
end

Binarization
Convert decimal to binary number in string
Syntax
Str= dec2bin(d)
str = dec2bin(d,n)
Description
returns the
str = dec2bin(d) binary representation of d as a string. d must be a nonnegative integer smaller than 2^52.
str = dec2bin(d,n) produces a binary representation with at least n bits.
Examples
Decimal 23 converts to binary 010111:
dec2bin(23)
ans = 10111
4.4. Binary to decimal

Convert binary number string to decimal number
Syntax
bin2dec(binarystr)
Description
bin2dec(binarystr) interprets the binary string binarystr and returns the equivalent decimal number.
bin2dec ignores any space (' ') characters in the input string.
Examples
Binary 010111 converts to decimal 23:
bin2dec('010111')
ans = 23
Because space characters are ignored, this string yields the same result:
bin2dec(' 010 111 ')
ans = 23

Linear polynomial Function

In linear polynomial function method, a polynomial function of order one is used to determine the byte of the cover medium where the secret data is going to be stored. The general format of the 1st degree polynomial function is Q = a.X + c. Based on this equation, the subsequent byte where the secret bit is to be kept is determined by

$$Xi+1 = (a.Xi + c) \bmod m$$

where a, c and m are constants. Xi determines the current byte position in the cover medium where the secret bit is stored and Xi+1 determines the next byte position where secret data can be stored. Assume that a = c = X1 = 3 and m= 5. And assume that the secret message that we need to hide is 101. And the source image data is

10100011 00001010 10001000
10000011 10101011 10100011
10001011 10011001 10100001
10101000 11101110 11110001

The first secret bit will be stored in X1 +1, which is the 4th position of the source file. So the 4th byte after inserting the secret bit will be 10000011. Now, the next bit of the secret data is to be stored based on the following calculation X2 = (a.X1 + c) mod 5. And it is 2 after calculation. So the next secret bit is to be stored in the byte position which is the sum of X1 + 1 and X2 +1 (addition of 1 is to get a unique cover medium byte since the above function can return 0) and it is 7th position. So the 7th byte after adding the secret bit is 10001010. Now the 3rd secret bit should be stored in the byte position which can be calculated based on X3 = (a.X2+ c) mod 5. And the value of X3 is 4. So the 3rd secret bit should be inserted in the 12th position which is X3 +1 position from the current position.

10100011 00001010 10001000
10000011 10101011 10100011
10001010 10011000 10100001
10101000 11101111 11110001

Quadratic polynomial function

In this method, a 2nd degree Polynomial function is used to determine the byte of the cover medium where the secret data is going to be stored. The general format of the quadratic polynomial function is Q = a.X2 + b.X + c. Based on this equation, the subsequent byte where the secret bit is to be kept is determined by

$$Xi+1 = (a.Xi2 + b.Xi + c) \bmod m$$

where a, b, c and m are constants. Xi determines the current byte position in the cover medium where the secret bit is stored and Xi+1 determines the next byte position where secret data can be stored.
Assume that a = b = c = X1 = 2 and m= 5. And assume that the secret message that we need to hide is 101. And the source image data is

10100011 00001010 10001000
10000011 10101011 10100011
10001011 10011001 10100001
10101000 11101110 11110001

The first secret bit will be stored in X1 +1, which is the 3rd position of the source file. So the 3rd byte after inserting the secret bit will be 10001001. Now, the next bit of the secret data is to be stored based on the following calculation X2 = (a.X12+ b.X1 + c) mod 5. After calculation X2 is 4. So the next secret bit is to be stored in the byte position which is the sum of X1 + 1 and X2+1 (addition of 1 is to get a unique cover medium byte since the above function can return 0) and it is 8th position. So the 8th byte after adding the secret bit is 10011000. Now the 3rd secret bit should be stored in the byte position which can be calculated based on X3 = (a.X3+ b.X2 + c) mod 5. And the value of X3 is 2. So the 3rd secret bit should be inserted in the 11th position which is X3 +1 position from the current position.

10100011 00001010 10001001
10000011 10101011 10100011
10001011 10011000 10100001
10101000 11101111 11110001

RESULTS

Measurement Analysis

As a performance measurement for image distortion, the well known Peak-Signal-to-Noise Ratio (PSNR) which is classified under the difference distortion metrics can be applied on the stego images. It is defined as:

$$PSNR = 10\log_{10}\left(\frac{C_{max}^2}{MSE}\right)$$

where MSE denotes Mean Square Error which is given as:

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy} - C_{xy}\right)^2$$

where x and y are the image coordinates, M and N are the dimensions of the image, S_{xy} is the generated stego-image and C_{xy} is the cover image. Also Cmax holds the maximum value in the image, for example:

$$C_{max}^2 \leq \begin{cases} 1, \text{double - precision} \\ 255, \text{uint8 - bit} \end{cases}$$

Consider Cmax=255 as a default value for 8-bit images. It can be the case, for instance, that the examined image has only up to 253 or fewer representations of gray colours. Knowing that Cmax is raised to a power of 2 results in a severe change to the PSNR value. Thus Cmax can be defined as the actual maximum value rather than the largest possible value. PSNR is often expressed on a logarithmic scale in decibels (dB). PSNR values falling below 30dB indicate a fairly low quality, i.e., distortion caused by embedding can be obvious; however, a high quality stego-image should strive for 40dB and above. Van Der Weken et al. proposed other similarity measures (SMs). They analysed the efficiency of ten SMs in addition to a modified version of PSNR constructed based on neighbourhood blocks which better adapt to human perception. In order to produce a fair performance comparison between different methods of invisible watermarking, Kutter and Petitcolas [86] discussed a novel measure adapted to the human visual system.

Peak signal-to-noise ratio (PSNR)

The phrase peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs it is used as an approximation to human perception of reconstruction quality, therefore in some cases one reconstruction may appear to be closer to the original than another, even though it has a lower PSNR (a higher PSNR would normally indicate that the reconstruction is of higher quality). One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.[1][2]

It is most easily defined via the mean squared error (MSE). Given a noise-free m×n monochrome image I and its noisy approximation K, MSE is defined as:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

The PSNR is defined as:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

Here, MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linearPCM with B bits per sample, MAX_I is 2^B-1. For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. Alternately, for color images the image is converted to a different color space and PSNR is reported against each channel of that color space, e.g., YCbCr orHSL.

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better. Acceptable values for wireless transmission quality loss are considered to be about 20 dB to 25 dB.

Mean squared error (MSE)

In statistics, the mean squared error (MSE) of an estimator is one of many ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the average of the squares of the "errors." The error is the amount by which the value implied by the estimator differs from the quantity to be estimated. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.[1]

The MSE is the second moment (about the origin) of the error, and thus incorporates both the variance of the estimator and its bias. For an unbiased estimator, the MSE is the variance of the estimator. Like the variance, MSE has the same units of measurement as the square of the quantity being estimated. In an analogy to standard deviation, taking the square root of MSE yields the root mean square

error or root mean square deviation (RMSE or RMSD), which has the same units as the quantity being estimated; for an unbiased estimator, the RMSE is the square root of the variance, known as the standard deviation.

The MSE of an estimator $\hat{\theta}$ with respect to the estimated parameter θ is defined as

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{E}\left[(\hat{\theta} - \theta)^2\right].$$

The MSE is equal to the sum of the variance and the squared bias of the estimator[2]

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + \left(\mathrm{Bias}(\hat{\theta}, \theta)\right)^2.$$

The MSE thus assesses the quality of an estimator in terms of its variation and unbiasedness. Note that the MSE is not equivalent to the expected value of the absolute error. Since MSE is an expectation, it is not a random variable. It may be a function of the unknown parameter θ, but it does not depend on any random quantities. However, when MSE is computed for a particular estimator of θ the true value of which is not known, it will be subject to estimation error. In a Bayesian sense, this means that there are cases in which it may be treated as a random variable.

In regression analysis, the term mean squared error is sometimes used to refer to the estimate of error variance: residual sum of squares divided by the number of degrees of freedom. This is an observed quantity given a particular sample (and hence is sample-dependent), whereas the definition above is a function of the parameters of the probability distribution of an unknown parameter. For more details, see errors and residuals in statistics.

Also in regression analysis, "mean squared error", often referred to as "out-of-sample mean squared error", can refer to the mean value of the squared deviations of the predictions from the true values, over an out-of-sample test space, generated by a model estimated over a particular sample space. This also is an observed quantity, and it varies by sample and by out-of-sample test space.
Suppose we have a random sample of size n from a population, $X_1, \ldots, X_n$. The usual estimator for the mean is the sample average

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n}(X_i)$$

which has an expected value of μ (so it is unbiased) and a mean square error of

$$\mathrm{MSE}(\overline{X}) = \mathrm{E}((\overline{X} - \mu)^2) = \left(\frac{\sigma}{\sqrt{n}}\right)^2 = \frac{\sigma^2}{n}$$

For a Gaussian distribution this is the best unbiased estimator (that is, it has the lowest MSE among all unbiased estimators), but not, say, for a uniform distribution. The usual estimator for the variance is

$$S_{n-1}^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} X_i^2 - n\overline{X}^2\right).$$

This is unbiased (its expected value is σ^2), and its MSE is[3]

$$\begin{aligned}\mathrm{MSE}(S_{n-1}^2) &= \frac{1}{n}\left(\mu_4 - \frac{n-3}{n-1}\sigma^4\right) \\ &= \frac{1}{n}\left(\gamma_2 + \frac{2n}{n-1}\right)\sigma^4,\end{aligned}$$

where μ_4 is the fourth central moment of the distribution or population and $\gamma_2 = \mu_4/\sigma^4 - 3$ is the excess kurtosis.
However, one can use other estimators for σ^2 which are proportional to S_{n-1}^2, and an appropriate choice can always give a lower mean square error. If we define

$$\begin{aligned}S_a^2 &= \frac{n-1}{a}S_{n-1}^2 \\ &= \frac{1}{a}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2\end{aligned}$$

then the MSE is

$$\begin{aligned}\mathrm{MSE}(S_a^2) &= \mathrm{E}\left(\left(\frac{n-1}{a}S_{n-1}^2 - \sigma^2\right)^2\right) \\ &= \frac{n-1}{na^2}[(n-1)\gamma_2 + n^2 + n]\sigma^4 - \frac{2(n-1)}{a}\sigma^4 + \sigma^4\end{aligned}$$

This is minimized when $a = \dfrac{(n-1)\gamma_2 + n^2 + n}{n} = n + 1 + \dfrac{n-1}{n}\gamma_2.$

Signal-to-noise ratio improvement (SNRI)

Signal-to-noise ratio (often abbreviated SNR or S/N) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. It is defined as the ratio of signal power to the noise power. A ratio higher than 1:1 indicates more signal than noise. While SNR is commonly quoted for electrical signals, it can be applied to any form of signal (such as isotope levels in an ice core or biochemical signaling between cells).The signal-to-noise ratio, the bandwidth, and the channel capacity of a communication channel are connected by the Shannon–Hartley theorem.

Signal-to-noise ratio is sometimes used informally to refer to the ratio of useful information to false or irrelevant data in a conversation or exchange. For example, in online discussion forums and other online communities, off-topic posts and spam are regarded as "noise" that interferes with the "signal" of appropriate discussion.Signal-to-noise ratio is defined as the power ratio between a signal (meaningful information) and the background noise (unwanted signal):

$$\mathrm{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}},$$

where P is average power. Both signal and noise power must be measured at the same or equivalent points in a system, and within the same system bandwidth. If the signal and the noise are measured across the same impedance, then the SNR can be obtained by calculating the square of the amplitude ratio:

$$\mathrm{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}}\right)^2,$$

where A is root mean square (RMS) amplitude (for example, RMS voltage). Because many signals have a very wide dynamic range, SNRs are often expressed using the logarithmic decibel scale. In decibels, the SNR is defined as

$$\mathrm{SNR}_{\text{dB}} = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right) = P_{\text{signal,dB}} - P_{\text{noise,dB}},$$

which may equivalently be written using amplitude ratios as

$$\mathrm{SNR}_{\text{dB}} = 10\log_{10}\left(\frac{A_{\text{signal}}}{A_{\text{noise}}}\right)^2 = 20\log_{10}\left(\frac{A_{\text{signal}}}{A_{\text{noise}}}\right).$$

The concepts of signal-to-noise ratio and dynamic range are closely related. Dynamic range measures the ratio between the strongest un-distorted signal on a channel and the minimum discernable signal, which for most purposes is the noise level. SNR measures the ratio between an arbitrary signal level (not necessarily the most powerful signal possible) and noise. Measuring signal-to-noise ratios requires the selection of a representative or reference signal. In audio engineering, the reference signal is usually a sine wave at a standardized nominal or alignment level, such as 1 kHz at +4 dBu (1.228 V_{RMS}).

SNR is usually taken to indicate an average signal-to-noise ratio, as it is possible that (near) instantaneous signal-to-noise ratios will be considerably different. The concept can be understood as normalizing the noise level to 1 (0 dB) and measuring how far the signal 'stands out. All real measurements are disturbed by noise. This includes electronic noise, but can also include external events that affect the measured phenomenon — wind, vibrations, gravitational attraction of the moon, variations of temperature, variations of humidity, etc., depending on what is measured and of the sensitivity of the device. It is often possible to reduce the noise by controlling the environment. Otherwise, when the characteristics of the noise are known and are different from the signals, it is possible to filter it or to process the signal.

Root-mean-square error (RMSE)

The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed. RMSD is a good measure of accuracy, but only to compare different forecasting errors within a dataset and not between different ones, as it is scale dependent. These individual differences are also called residuals, and the RMSD serves to aggregate them into a single measure of predictive power. The RMSD of an estimator $\hat{\theta}$ with respect to the estimated parameter θ is defined as the square root of the mean square error:

$$\mathrm{RMSD}(\hat{\theta}) = \sqrt{\mathrm{MSE}(\hat{\theta})} = \sqrt{\mathrm{E}((\hat{\theta} - \theta)^2)}.$$

For an unbiased estimator, the RMSD is the square root of the variance, known as the standard error.In some disciplines, the RMSD is used to compare differences between two things that may vary, neither of which is accepted as the "standard". For example, when measuring the average distance between two oblong objects, expressed as random vectors

$$\theta_1 = \begin{bmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,n} \end{bmatrix} \quad \text{and} \quad \theta_2 = \begin{bmatrix} x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{2,n} \end{bmatrix}.$$

The formula becomes:

$$\mathrm{RMSD}(\theta_1, \theta_2) = \sqrt{\mathrm{MSE}(\theta_1, \theta_2)} = \sqrt{\mathrm{E}((\theta_1 - \theta_2)^2)} = \sqrt{\frac{\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2}{n}}.$$

Normalized root-mean-square deviation (NMSE)

The normalized root-mean-square deviation or error (NRMSD or NRMSE) is the RMSD divided by the range of observed values,[citation needed] or:

$$\mathrm{NRMSD} = \frac{\mathrm{RMSD}}{x_{\max} - x_{\min}}$$

The value is often expressed as a percentage, where lower values indicate less residual variance.

III. CONCLUSION

A detailed overview is given on the steganography techniques in the beginning of the paper. After that, a detailed explanation has been given for the newly proposed steganographic algorithms based on polynomial functions and their advantages over the existing method. Also discussion has been carried out to illustrate the new algorithms with examples and it concludes that the proposed algorithms which select the cover medium bytes randomly to store the secret text is better and stronger than the conventional steganographic techniques.

REFERENCES

[1]. F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding - a Survey," Proceedings of the IEEE, Vol. 87, pp. 1062–1078, 1999.

[2]. W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for data hiding," IBM Systems Journal Vol. 35 (3–4), pp. 313–336, 1996.

[3]. Y. K. Lee, L. H. Chen, "High capacity image steganographic model," IEE Proceedings on Vision, Image and Signal Processing, Vol. 147, No.3, pp. 288-294, 2000.

[4]. R.-Z. Wang, C.-F.Lin, and J.-C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," Pattern Recognition Vol. 34, pp. 671–683, 2001.

[5]. C.-C. Chang, J.-Y.Hsiao, C.-S.Chan, "Finding optimal least significant-bit substitution in image hiding by dynamic programming strategy," Pattern Recognition Vol.36, Issue 7, pp. 1583-1595, 2003.

[6]. C.-K. Chan, L. M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recognition Vol. 37, Issue 3, pp. 469-474, 2004.Issue 3, pp. 469-474, 2004.

[7]. W.-N. Lie, and L.-C. Chang, "Data hiding in images with adaptive numbers of least significant bits based on the human visual system," IEEE International Conference on Image Processing, Vol. 1, pp. 286–290, 1999.

[8]. D.-C. Wu, and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," Pattern Recognition Letters, Vol. 24, pp. 1613–1626, 2003.

[9]. H.-C. Wu, N.-I.Wu, C.-S.Tsai, and M.-S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," IEE Proceedings on Vision, Image and Signal Processing, Vol. 152, No. 5, pp. 611-615, 2005.

[10]. S.-L. Li, K.-C.Leung, L.-M.Cheng, and C.-K. Chan, "Data Hiding in Images by Adaptive LSB Substitution Based on the Pixel-Value Differencing," First International Conference on Innovative Computing, Information and Control (ICICIC'06), Vol. 3, pp. 58-61, 2006.

[11]. [11] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks," IEEE Communications Magazine, Vol. 39, Issue 8, 118-126, 2001.

[12]. [12] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color, and gray-scale images," IEEE multimedia, Vol. 8, Issue 4, pp. 22-28, 2001.