



اخفاء رسالة نصية في فيديو باستخدام طريقة الوحدة الثنائية الاقل دلالة
المحسنة

HIDING TEXT MESSAGE IN VIDEO USING IMPROVED LSB METHOD

By

Maher Mohammed Al-Essa

Supervisor

Prof. Dr. Alaa Al-Hamami

A Thesis Submitted

In Partial Fulfillment of the Requirements for the Degree

Of Master in Computer Science

Department of Computer Science

College of Computer Sciences and Informatics

Amman Arab University

July 2014

تفويض



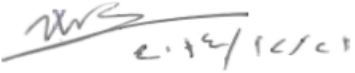
أنا ماهر محمد العيسى أقوض جامعة عمان العربية بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات أو الأشخاص عند طلبهم حسب التعليمات النافذة في الجامعة.

الإسم: ماهر محمد العيسى

التوقيع: 

التاريخ: ١٢ / ١٤ / ٢٠١٤

This thesis titled “Hiding text message in video using improved LSB methods” submitted by Maher M. Al-Essa was examined and approved on date 21 / 12 / 2014 by the examining committee as follows:

<u>التوقيع و التاريخ</u>	<u>الإسم الثلاثي</u>
	1. أ.د. علاء حسين الحمامي (رئيسا / مشرفا)
	2. أ.د. عاصم الشيخ (عضوا / خارجيا)
	3. د. دفينوس سماوي (عضوا)

Dedication

To my family

For their help and support

Acknowledgment

*I would like to thank my supervisor Prof. Dr. Alaa Al-Hamami
for his support and guidance and all his help to complete this
achievement*

I also would like to thank Amman Arab University

Hiding Text Message in Video using improved LSB Method

Abstract

This work aims at focusing or improving Least Significant Bit (LSB) method using measure the standard deviation, by answering the following questions: How to hide text in video based on the Least Significant Bit? And the sub question is: What are the steps to illustrate procedure of preparing video to hide text message in the LSB?

The study suggests algorithm concealment improved transfer confidential letter to the formula of (Binary) , followed by the division of the image to hide data where clips blocks size (32×32) and account values standard deviation (Standard Deviation “SD”) for each section are then finding minimum and the maximum value of a standard deviation in addition to the average value, then isolate sections where the value of the standard deviation less than average value to be key concealment (i.e be adopted as locations to hide) and that by including all bits of the message into the (LSB) for each section of the selected sections, and the research used (Microsoft .Net C#).

The results proved the efficiency of the algorithm, where the hidden information did not cause any distortion on the video cover used. The study suggested a future work in using an innovative way to compress the video output to return it to the original size without losing some of the data.

إخفاء رسالة نصية في فيديو باستخدام طريقة الوحدة الثنائية الأقل دلالة المحسنة

الملخص

تهدف الدراسة إلى التركيز على تحسين تقنية LSB باستخدام قياس الانحراف المعياري، من خلال الإجابة على الأسئلة التالية: كيفية إخفاء النص في الفيديو على أساس الوحدة الثنائية الأقل دلالة؟ وما هي الخطوات المطلوبة لتوضيح إجراءات إعداد الفيديو لإخفاء رسالة نصية في LSB؟

اقترحت الدراسة خوارزمية الإخفاء لتحسين نقل رسالة سرية إلى صيغة (ثنائي)، تتبعها تقسيم الصورة بحيث تكون البيانات المراد إخفاؤها في كتل مقاطع حجم (32×32) وحساب قيم الانحراف المعياري (SD) لكل قسم. ثم يتم تحديد أقل وأكبر قيمة انحراف معياري بالإضافة إلى المتوسط الحسابي، ثم عزل الأقسام ذات الانحراف المعياري الأقل من المتوسط الحسابي لتكون مكان الإخفاء الرئيسي (أي اعتمادها كمواقع للإخفاء) وتضمن كافة البايتات في (LSB) لكل قسم من الأقسام المختارة، وقد استخدمت الدراسة (مايكروسوفت دوت نت #C).

وأثبتت نتائج الدراسة كفاءة الخوارزمية، حيث أن المعلومات المخفية لم تحدث أي تشويه لها على غلاف الفيديو المستخدم، وأظهرت النتائج أيضا أن تقسيم الصور أدى إلى زيادة في مساحة التخزين الممكن استخدامها لتخزين النص السري.

واقترحت الدراسة كعمل مستقبلي، استخدام طريقة مبتكرة لضغط الفيديو الناتج لإعادته إلى حجمه الأصلي دون أن يفقد بعض البيانات.

List of Abbreviations

ASCII	American Standard Code for Information Interchange
AVI	Audio Video Interleave
DVD	Digital Versatile Disk
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
MKV	Matroska Video
RAW	Raw Image Format
RGB	Red Green & Blue
SD	Standard Deviation
WMV	Windows Media Video

List of Figures

Figure	Content	Page
1-1	Distribution of light and dark pixels	5
2-1	Steganographic System	14
3-1	Proposed Model	29
3-2	Hiding Procedure	31
3-3	Hide-LSB	33
3-4	Receiver Procedure	35
3-5	LSB Extraction	38
4-1	Steps of Model	43
4-2	Application Interface	44
4-3	Select Cover Video	45
4-4	Select Secret	45
4-5	Hidden Progress	46
4-6 A	Figure (4-6 A)	47
4-6 B	Figure (4-6 B)	47
4-7	distribution of (RGB) color	48
4-8	Used Blocks	49
4-9	Frame Blocks Division	51
4-10	Frame Block	51
4-11	Chosen Frame Block	53
4-12	Select Stego Video	55
4-13	Extraction Progress	55
4-14	Write Detailed Data	58

Table of Contents

Authorization.....	b
Committee Decision	c
Dedication	d
Acknowledgment	e
Abstract	f
الملخص	g
List of Abbreviations.....	h
List of Figures	i
Table of Contents	j
Chapter I Introduction	2
1.1 Introduction.....	2
1.2 Objectives of Research	6
1.3 Statement of Problem	7
1-3-1 Hiding procedure	8
1.3.2 Extraction procedure.....	8
1.4 Importance of Research	8
1.5 Thesis Organization	9
Chapter two Literature Review	11
Section 1: Theoretical Framework	11
2.1 Overview	11
2.2 Steganography.....	15
2.3 Injection (or insertion).....	16
2.4 Substitution	17
2.5 Generation	18
2.6 Steganography vs. Cryptography	18
Section 2: Literature Review	20
Chapter three Conceptual Design of the Research	27
3.1 Methodology of Research	27

3.2 Introduction.....	28
3.3 Hiding procedure.....	29
3.4 Extraction procedure.....	30
3.5 Hiding procedure:.....	32
3.6 Hiding Algorithm:	32
3.7 Algorithm: Hiding algorithm:-	33
3.8 Secret and cover video file identification:.....	34
Create stego video file:	34
LSB Hiding Algorithm.....	35
Algorithm: LSB Hiding algorithm	36
Save Stego Video	37
3.9 Receiver procedure	37
Receiver Algorithm:.....	38
Algorithm: Receiver algorithm:-	38
Extract stego video	39
LSB Extraction Algorithm.....	40
Algorithm: LSB Extraction algorithm	41
Retrieve secret text	42
Limitation	42
Chapter four The Experimental Work	45
4.1 Preface:.....	45
4.2 Sender part:.....	47
Example:	54
4.3 Receiver part:	58
Example :	60

Hiding Detailed Example:	62
Chapter five Conclusion.....	67
5-1 Conclusion:.....	67
5-2 Future works:	68
References:.....	69

Chapter one

Introduction

Chapter I

Introduction

1.1 Introduction

In today's society the most practical implementation of steganography is used in the world of computers. Data is the heart of computer communication and over years a lot of methods have been developed to accomplish the goal of using Steganography to hide data. The trick is to embed an object within significantly larger objects, so the change is undetectable by the human eye.

In computing, the Least Significant Bit (LSB) is the bit position in a binary integer giving the units value, that is, determining whether the number is even or odd. The (LSB) , is the right-most bit due to the convention in positional notation of writing least significant digit further to the right. It is analogous to the least significant digit of a decimal integer, which is the digit in the ones (right-most) position.

The concealment is the most important means used by the security institutions with critical communications in all countries of the world. They provided the technology of high security, especially in the communication networks and the Internet.

The research will use algorithm concealment to improve transfer confidential letter to the formula of (Binary), followed by the division of the image to be hide data where clips blocks size is (32×32) and find the Standard Deviation (SD) for each block . Then find the minimum and the maximum value of a standard deviation in addition to the average value. Isolate blocks where the value of the standard deviation less the average value to be key concealment and that by including all bit of message into the LSB in each blocks of the selected blocks.

Since the LSB will provide a better results, for example the hidden information will not cause any distortion on the cover files, also the dividing of the image to cover a range of sections will lead to increasing the strength of the improved algorithm, on the other hand, the blocks with the less standard deviation have a less dispersion of data, while the standard deviation measures the amount of variation or dispersion from the average (Bland & Altman, 1996).

According to that, hiding text inside an image within several images are difficult to predict by the intercept party and to find that there is a message within the image, and it will also be difficult to resolve the hidden text.

For this reason this research relate with video to hide information, because video is a sequence of frames and each frame is an image, then there will be many images on the same file, which increase the size of hidden information and make it more complex than on image embedded data.

Furthermore, we use compression to make files smaller, allowing them to download faster and take up less storage space. For example, when we take a photo, the camera captures all the light it can get and puts together an image. If we save the image in Raw Image Format (RAW) format, which keeps all the light data the camera's sensor received, the image may be as large as 25 MB (it depends on the resolution of the image, a camera with more megapixels will produce a larger image). Note that lossy formats generally have a setting that controls how lossy they are. For example, Joint Photographic Experts Group (JPEG) has a variable quality setting. Low quality makes a smaller JPEG image file, but the quality of the image is noticeably worse (Mathkour, et al., 2008).

Now, in the case of this research it deals with video, so we have the Audio Video Interleave (AVI) as a lossless video formats in common consumer use, as they would result in video files taking up a huge amount of space.

Common formats like H.264, Matroska Video (MKV), and Windows Media Video (WMV) are all lossy. H.264 can provide smaller files with higher quality than previous generations of video codecs because it has a “smarter” algorithm that is better at choosing the data to throw out.

In addition there is some attackers who are using several methods to detect if there is any confidential information hidden or not, one of these methods is the use of the histogram to compare one image to another by distribution RGB colors.

A histogram is a way to graphically represent the distribution of data in a data set. Each data point is placed into a bin based on its value. The histogram is a plot of the number of data points in each bin. In scientific experiments, histograms are useful in characterizing the spread of data from repeated trials and for determining the probability of given measurement.

Digital image histograms are presented as a bar chart with the horizontal axis being the tonal range of your image, the left side of the graph shows how many dark pixels you have in your image, while the right side of the graph shows how many light pixels you have in your image, which the histogram shows you the distribution of light and dark pixels in your image as in Figure 1-1.

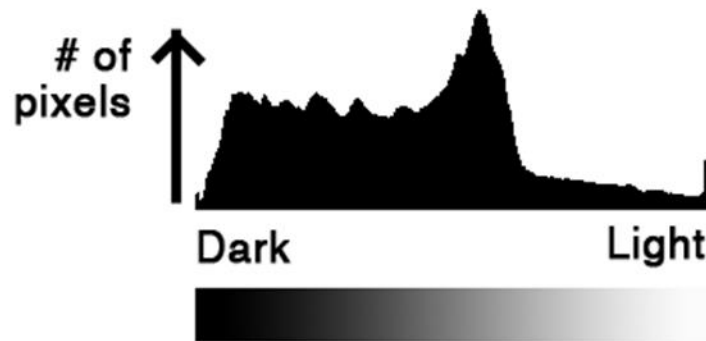


Figure (1-1) : Distribution of light and dark pixels
According to that, this research is trying to focus on improving LSB

method using measurement of the standard deviation, by hiding information within a video file in order to safely deliver the information to the other party, and by examining how it is possible the eligibility of the video through inserting a watermark within the file to prove the eligibility of it.

1.2 Objectives of Research

The objective of this research is to transfer a secret message from one side to another. The secret message will be hidden into video, due to the large size of the video. The hidden message will not make any difference in the statistical features of the original video.

This method is used for authentication of originality of the video. Video is a sequence of frames and each frame is an image, then there will be many images on the same file, which increase the size of hidden information and make it more complex than on image embedded data.

1.3 Statement of Problem

Data Hiding is the art and science of writing hidden messages in such a way that no one apart from the sender and intended recipient even realizes there is a hidden message. Nowadays, were the news and world events that occur are photographed directly through cameras or mobiles and send directly to all parts of the world where that information is worth their time, and to maintain the intellectual property and the rights of photographers and journalists and to prevent manipulation of entitlement, there are several ways to do this as in watermark and information hiding. In this research I choose a simple and fast process of concealment and retrieve data in a manner of hiding text message in video using improved LSB methods, as following:

1-3-1 Hiding procedure

The hiding procedure starts with choosing a cover video to encrypt the secret text in it, and decode it into frames with identical height and width. After that calculating the average standard deviation for all the blocks in that frame and filtering out the blocks with a standard deviation greater than the calculated average standard deviation for all the blocks. Finally, hiding a chunk of binary data, will take place, in each of the filtered blocks by using LSB.

1.3.2 Extraction procedure

This stage will start with downloading or receiving the video file, and decoding it and to extract the frames, then dividing each of the extracted frames into pixels blocks and finding the block with the least standard deviation. Then store all the extracted data in a file.

1.4 Importance of Research

The importance of this research showed in hiding information within a video file in order to safely deliver the information to the other party. On the other hand, it is possible the eligibility of the video through inserting a watermark within the file to prove the eligibility of it.

1.5 Thesis Organization

In addition to chapter one, this thesis contains four more chapters. Chapter two discusses the literature reviews and chapter three discusses the conceptual design of the research study. Chapter four explains the experimental works and finally chapter five will give the conclusions and future works of the research.

Chapter two

Literature Review

Chapter two

Literature Review

This chapter is divided into two sections, the first section deals with theoretical framework of the research, where the second section is the literature review, as following:

Section 1: Theoretical Framework

The purpose of this section is to discuss the most important topic in information age, which is information security, where the research will discuss the Least Significant Bit (LSB), in addition to steganography and other security methods such as injection, substitution and generation, as following:

2.1 Overview

A majority of the messages hidden today are hidden inside digital images, audio files or video files. But even modern printers can hide messages with the way they print the text out. For example, laser printers are so precise; they can offset a letter by $1/300^{\text{th}}$ of an inch. By doing this at certain points, they could send a binary message, which would be undetectable to the naked eye. The way it works is that a normal space would be considered a “0” while spaces that are offset by $1/300^{\text{th}}$ of an inch would be considered the “1’s”

This is good for hiding messages in print form, but has not solved the problem of sending the hidden message from computer to computer. This is where the files come into play (Sharma & Shrivastava, 2012).

Hidden files or pictures can be hidden in picture files because pictures files are so complex. Pictures on a computer are represented by tons and tons of pixels. Each pixel consists of a variation of all three primary colors, red, green and blue. In a standard 24-bit bitmap, 8 bits will represent each of the three colors. 8 times 3 is 24. That means there are 256 different variations of each color in every pixel that makes up a picture. So, to represent the color white, the code would look like 11111111 11111111 11111111. Now, the human eye cannot distinguish the difference between too many colors and so the color 11111110 11111110 11111110 would look exactly the same as white. Because of this, the last digit in every bit in every pixel could be changed. This is the basis of the Least Significant Bit Insertion technique (Chen & Lin, 2006).

Now, to explain the idea, you only need 8 bits to represent ASCII text and there are three extra in every pixel of a picture. Therefore, with every three pixels, you could form one letter of ASCII text. This may not seem like a lot, but when the standard image size is 640×480 pixels, that add up to a lot in a hurry. In order to make this practical to the user,

a computer program would be needed. After you type in your secret message and determine a cover message (the picture you want to hide your message in) the program would go through every pixel and change the last digit to represent each letter of the message you wrote. You would then send the picture to the correct recipient who would then use his program to go through every pixel and take off the last digit and use that to form the message (Mathkour & Al-Sadoon, 2008).

The problem of using steganography over digital communications has been solved. Also, the great thing about LSB (Least Significant Bit Insertion) is that the message is not lost if the file is compressed. Anyone who uses online pictures knows that bitmap files hold a lot of information and so are generally large in size. But because the secret message is encoded into the color bits, the message is never lost when compressed. The one problem with this approach is that it does not work for every picture type. LSB works mainly with Bitmaps because of the way bitmaps are compressed. JPEG's, on the other hand, are compressed using sophisticated algorithms and so a lot of the original information is lost (Thota & Devireddy, 2008).

Because information could so easily be lost with certain compression programs, other techniques were developed.

One technique is called the Masking and Filtering technique. This technique is very similar to watermarking. The image is marked with the secret message or image and then cannot be seen unless the luminosity level is changed to an exact amount. This worked better because the text/image was now actually part of the picture and no longer in the coding part. Another technique developed used the way certain pictures are compressed to its advantage. As stated earlier, JPEG's are compressed using sophisticated algorithms and because of this, a lot of the original information of the picture is lost. So, basically, what this last technique does is, it determines how the picture is going to be compressed with all the algorithms. It then changes the information of the picture accordingly to the secret message. It changes the information in a way that when decompressed, it will look similar to the LSB approach. This way, when the picture is viewed, it still looks the same but the secret message could be determined by taking the last bit of each pixel just like the LSB approach (Walia & Navdeep, 2010).

2.2 Steganography

In a digital image, information can be inserted directly into every bit of image information or the more busy areas of an image can be calculated so as to hide such messages in less perceptible parts of an image (Lee et al., 2008; Titty, 2009).

Tirkel et al (1993) was the first one who used techniques for image watermarking. Two techniques were presented to hide data in the spatial domain of images by them. These methods were based on the pixel value's Least Significant Bit (LSB) modifications. The algorithm proposed by Kurah and McHughes (1992) to embed in the LSB and it was known as image downgrading (Zheng et al., 2007).

Steganography is the art and science of writing hidden messages inside innocent looking containers such as digital files, in such a way that no one apart from the sender and intended recipient realizes the existence of a hidden message (Alalem & Manasrah, 2008). Steganography uses redundant portions of the container file such as Video files to embed the secret message.

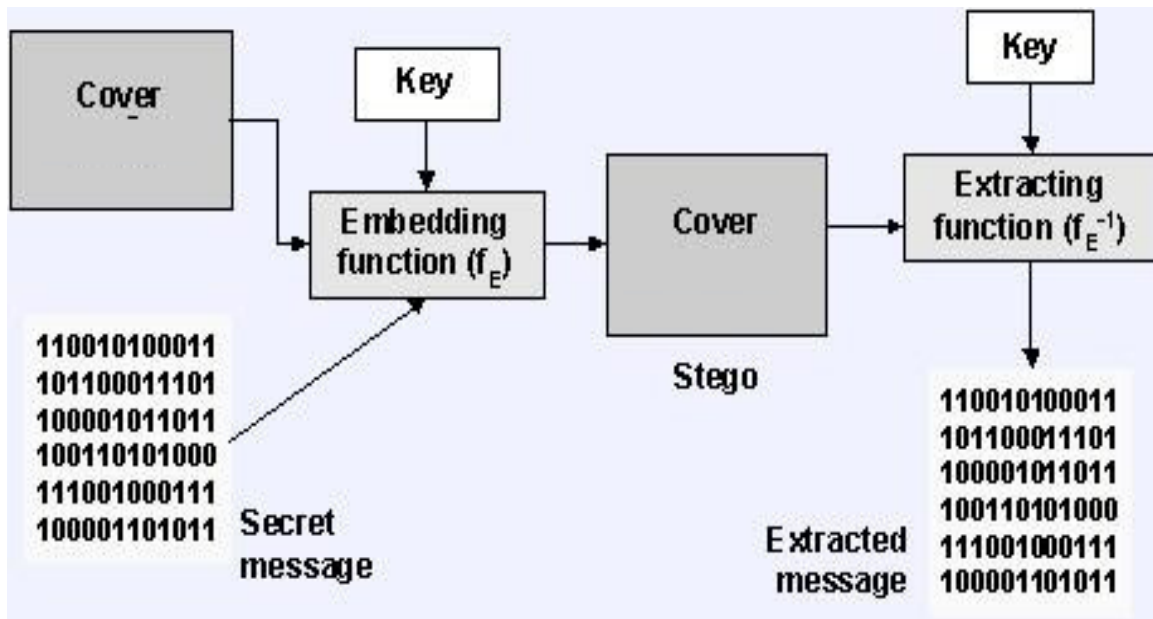


Figure (2-1): Steganographic System (Krenn, 2004)

Figure 2-1 gives an overview of the Steganographic system. There are three different types of Steganographic algorithms namely Injection, Substitution and Generation.

2.3 Injection (or insertion)

This technique adds bits to unused sections of digital files to hide the secret message. By doing this we avoid modifying those file bits that are relevant to an end-user—leaving the cover file perfectly usable. For example, we can add additional harmless bytes in an executable or binary file. Because those bytes do not affect the process,

the end-user may not even realize that the file contains additional hidden information. Using an insertion technique changes file size (Johnson & Jajodia, 1998).

2.4 Substitution

This technique is used to replace the least significant bits of information that determine the meaningful content of the original file with new data in a way that causes the least amount of distortion. The main advantage of this technique is that the cover file size does not change after the execution of the algorithm. On the other hand, this approach has few drawbacks. The resulting stego-file may be adversely affected by quality degradation and that may raise suspicion. Another drawback is substitution method limits the amount of data that can be hidden to the number of insignificant bits (Wang & Wang, 2004).

. Among the substitution techniques, a very popular methodology is the LSB (Least Significant Bit) algorithm, which replaces the least significant bit in some bytes of the cover file to hide a sequence of bytes containing the hidden data. That is usually, an effective technique in cases where the LSB substitution does not cause significant quality degradation (such as in 24-bit bitmaps) (Kumar & Mahesh, 2007).

2.5 Generation

Unlike injection and substitution, this technique does not require an existing cover file. This technique generates a cover file for the sole purpose of hiding the message. The main flaw of the insertion and substitution techniques is that people can compare the stego file with any pre-existing copy of the cover file (which is supposed to be the same file) and discover differences between the two. We will not have that problem when using a generation approach, because the result is an original file, and is therefore immune to comparison tests

2.6 Steganography vs. Cryptography

Steganography and Cryptography are parallel data security techniques, both can be implemented side by side but, they differ in certain qualities like (Dennis et al., 2004):

1. Steganography can use cryptography but not vice versa.
2. Steganography has a very expensive payload as compared to cryptography.

3. Cryptography makes the message “unreadable” where as Steganography makes it “unseen”.

Steganography implemented to cryptographic data will increase the security of the data communication.

Still, 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 (Cole, 2003).

Finally; Steganography is also being used everyday life for practical needs. Odds are, you encounter the use at least once a week and do not even know it. One of the biggest uses today is with copyrighted materials like DVDs. DVDs are actually encoded with certain watermarks that the DVD player recognizes. The watermark has numerous functions.

First it tells where the DVD came from so if someone makes copies of their DVDs, the original copy could always be determined. Secondly, the watermark determines if the DVD could actually be copied or not. Finally, the watermark tells the DVD player if it could play the DVD or not. Unknown to a lot of people, but DVDs are made in certain “Regions” and they only work in that region. For example, Asia and North America are considered different regions and so a DVD from Asia will not play on a DVD player that has a North America region code.

Seeing how complex steganography is today, it is hard to imagine what the future could hold. How're as technology is growing exponentially, the bounds for steganography seem limitless. One day, hiding a message inside someone's brain without the person even knowing it, Johnny Mnemonic style, may become a reality.

Section 2: Literature Review

This section deals with the studies that reviewed the subject of the current research, and the studies were arranged in descending order from newest to oldest, as follows:

Sharma & Shrivastava (2012) “A Steganography Algorithm for Hiding Images by improved LSB substitution by minimize detection”.

The study has worked upon a new steganography algorithm for 8bit (gray scale) or 24bit (color image) based on Logical operation to ensure the security against the steganalysis attack.

Ramalingam (2011) “Stego Machine – Video Steganography using Modified LSB Algorithm”.

This study designed a stego machine to develop a steganographic application to hide data containing text in a computer video file and to retrieve the hidden information. It was designed by embedding text file in a video file in such way that the video does not lose its functionality using Least Significant Bit (LSB) modification method. This method applies imperceptible modifications. This proposed method strives for high security to an eavesdropper’s inability to detect hidden information.

Al-Neamah & Al-Neamah (2010), “Design and Implementation of Steganographic Algorithm on Video File (mov)”.

In this research, video file type (mov) was used to hide an English text. This method offered high accuracy and secure for data transitions.

First, frames are extracted from video file. Then an algorithm has been designed and implemented to hide and extract the text message. Hiding process concerns with converting the text into corresponding codes, then store these codes inside the basic color panel of video file and exactly on the fourth order after floating point of every pixel. Extracting process concerns with inverting the whole process of hiding. Experimental results demonstrated success of hiding process. The study used Matlab (ver. 7).

Jalab, et al., (2009) “Frame Selected Approach for Hiding Data within MPEG Video Using Bit Plane Complexity Segmentation”.

The study proposed a collaborate approach for selecting frame for Hiding Data within MPEG Video Using Bit Plane Complexity Segmentation. This approach invented high secure data hidden using select frame from MPEG Video, and furthermore assigned the well-built of the approach.

In addition to the security issues, the study used the digital video as a cover to the hidden data. The reason behind opt the video cover in this approach is the huge amount of single frames image per sec which in turn overcome the problem of the data hiding quantity, as the experiment result showed the success of the hidden data within select frame, extract data from the frames sequence. These function without affecting the quality of the video.

Beenish & Faruqui (2008), “A Steganography Implementation”.

The study discusses the art and science of Steganography in general and proposes a novel technique to hide data in a colorful image using least significant bit.

Mathkour, et al., (2008), “A New Image Steganography Technique”.

The study set criteria to analyze and evaluate the strengths and weaknesses of the presented techniques and a more robust steganography technique has been developed that takes advantage of the strengths and avoids the limitations.

Jian & Gupta (2007), “A JPEG Compression Resistant Steganography Scheme for Raster Graphics Images”.

The researchers proposed a scheme which hides data in bitmap images, in a way that there is almost no perceptible difference between the original image and this new image and which is also resistant to JPEG compression.

Po-Yueh & Hung-Ju (2006), “A DWT Based Approach for Image Steganography”.

The study proposed a steganography technique which embeds the secret messages in frequency domain, according to different users' demands on the embedding capacity and image quality, the proposed algorithm is divided into two modes and 5 cases.

Chen, et al., (2006), “Analysis of Current Steganography Tools: Classifications & Features”.

Their study focused on the steganography tools algorithms, based on the analyses of the algorithms, various tools are divided into five categories: (a). Spatial domain based steganography tools; (b). Transform domain based steganography tools; (c). Document based steganography tools; (d) File structure based Steganography tools; (e) other categories, e.g. video compress encoding and spread spectrum technique based.

Raja et al., (2005), “A Secure Image Steganography using LSB, DCT and Compression Techniques on Raw Images”.

The study proposed a challenging task of transferring the embedded information to the destination without being detected; the paper showed that the image based steganography that combines Least Significant Bit (LSB), Discrete Cosine Transform (DCT), and compression techniques on raw images to enhance the security of the payload.

Deshpande & KamalapurSnehal (2004), “Implementation of LSB Steganography and Its Evaluation for Various Bits”.

The study proposed the Least Significant Bit embedding technique suggests that data can be hidden in the least significant bits of the cover image

and the human eye would be unable to notice the hidden image in the cover file, the paper explains the LSB embedding technique and presents the evaluation results for 2, 4, 6 Least significant bits for a .png file and a .bmp file.

The differences between the current study and previous studies:

According to the review of the previous studies, it's clear that the literature review of previous studies showed that it dealt with the experiences of different attempts about hiding text inside the videos, using a different programming languages, but the current study addresses to hide text inside video using the LSB technique by programming with Microsoft .Net C#, which is not exposed by any previous studies.

Chapter three

Conceptual Design of the Research

Chapter three

Conceptual Design of the Research

This chapter is about the methodology used in this research, plus the algorithms and programming languages used in the research, as following:

3.1 Methodology of Research

The methodology of the research is based on theoretical and applied methodology, where the theoretical methodology deals with literature review and references about hiding text inside videos with LSB, while the applied methodology dealing with practical application of this process by conducting practical experience shows these results.

The research suggests algorithm concealment that improved transfer confidential letter to the formula of (Binary) followed by the division of the image to hide data where clips blocks size (32×32) and account values standard deviation (Standard Deviation “SD”) for each section are then finding minimum and maximum value of a standard deviation in addition to the average value,

then isolate sections where the value of the standard deviation less the average value to be key concealment (i.e be adopted as locations to hide) and that by including all bit of message into the (LSB) for each section of the selected sections, and the research used (Microsoft .Net C#).

3.2 Introduction

As in Naforita et al., (2006) the researchers approved that using LSB based on standard deviation will provide better results, for example the hidden information will not cause any distortion on the cover files. Dividing of the image to cover a range of blocks will lead to increase the strength of the improved algorithm; on the other hand, the blocks with the less SD or equal have a less dispersion of data.

According to that, hiding text inside an image within several images are difficult to predict by the intercept party and to find that there is a message within the image, and it will also be difficult to resolve the hidden text.

For this reason this research relate with video to hide information, because video is a sequence of frames and each frame is an image, then there will be many images on the same file, which increase the size of hidden information and make it more complex than on image embedded data.

According to that, this research is trying to focus on improving LSB method using measure the standard deviation.

The proposed model includes two procedures, sender and receiver, as follow:

3.3 Hiding procedure

This section provides the steps were followed by the researcher in the first stage of hiding procedure:

1. A cover video file is chosen to hide the secret text in it, and then the secret text is prepared for hiding by converting it from American Standard Code for Information Interchange (ASCII) to binary system.
2. The cover file will be decoded into frames with identical height and width.
3. Filtering the frames images, so that the secret text will not be hiding in consecutive frames, this will make resolving secret text from the video more difficult for any intercepting parties. At this stage we have the binary representation of the secret text and the frames from the video in which the binary data will be hidden in it.
4. Each frame from the filtered frames will be divided into 32×32 pixels blocks and calculating the standard deviation for each block.

5. Calculating the average standard deviation for all the blocks in that frame and filtering out the blocks with a standard deviation greater than the calculated average standard deviation for all the blocks.
6. hiding a chunk of the binary data in each of the filtered blocks by using LSB. After hiding all the binary data, the resultant stego frames will be converted back to video; the result will be a stego video file that can be sent to the receiving party.

3.4 Extraction procedure

This section provides the steps of retrieving the secret text :

1. Downloading or receiving the video file.
2. The video file will be decoded to extract the frames.
3. Divided each of the extracted frames into 32×32 pixels blocks and find the standard deviation for each block. Find the map block, the map block is the block with the minimum standard deviation.
4. Traversing the blocks that has data in it and extracting the data from it, the map block tells where if the block has data or not.
5. Show all the extracted data.

Figure (3-1) shows each procedure in the proposed model.

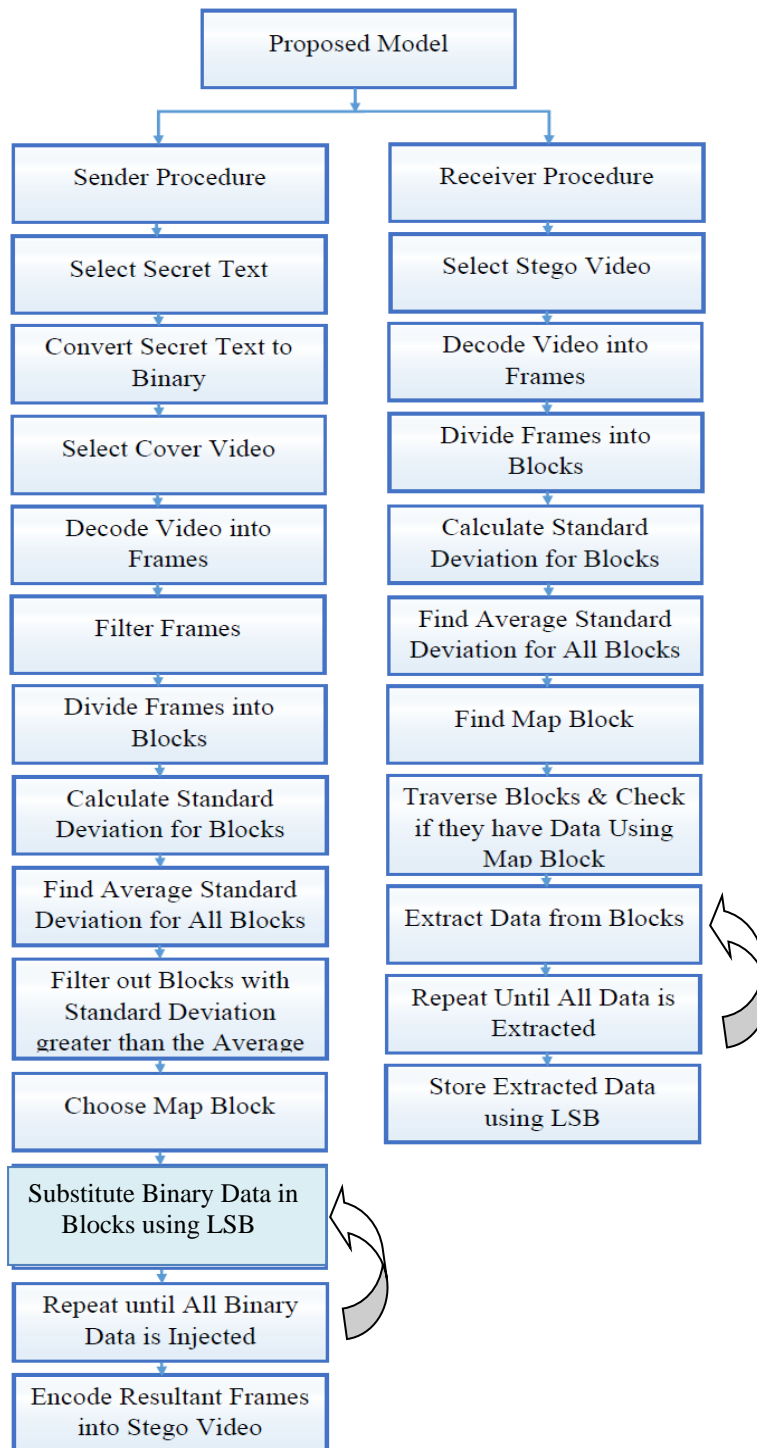


Figure (3-1) ... Proposed Model

3.5 Hiding procedure:

The hiding procedure includes the steps that will hide the secret text in the cover video file (stego video). The hiding procedure steps starts from getting the secret text along with the video cover file and ends with saving a video that contains the secret text (resultant stego video file). The hiding procedure consists of the following steps:

1. Enter secret text
2. Select cover video file
3. Hiding secret text in the cover video using LSB
4. Saving stego video file.

3.6 Hiding Algorithm:

The hiding algorithm purpose is creating a stego video file that includes a secret text hidden within it in a way that is very difficult for any intercepting party to identify.

The hidden algorithm works as shown in Figure (3-2):-

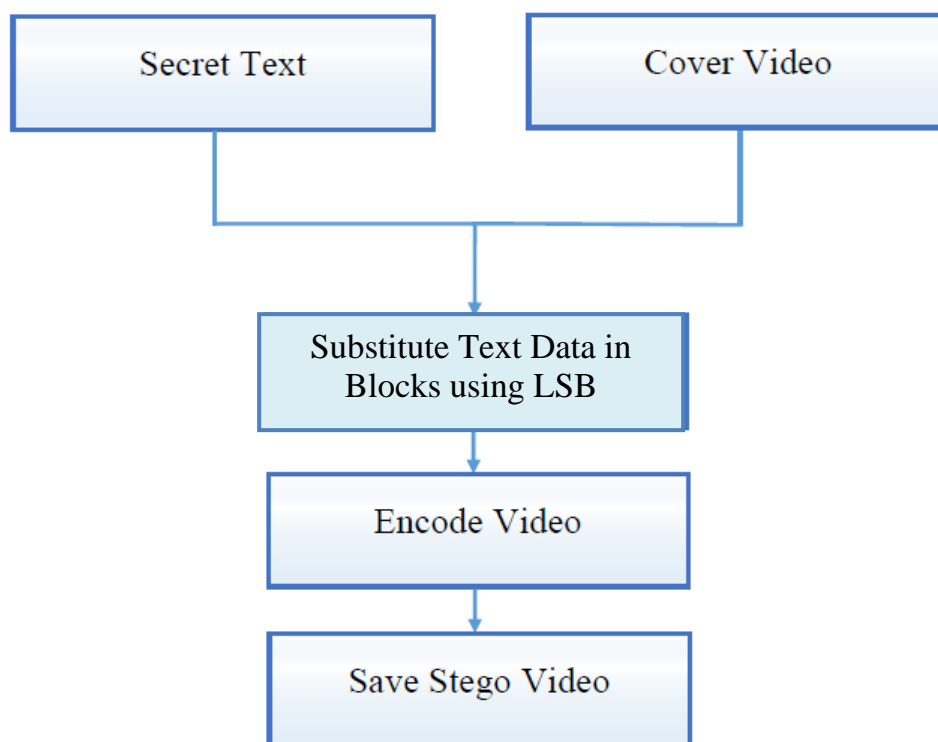


Figure (3-2) ... Hiding Procedure

3.7 Algorithm: Hiding algorithm:-

// Input algorithm: - secret text and cover video file.

// Output algorithm: - Stego video file.

- $X \leftarrow$ Secret text.
- $Y \leftarrow$ Cover AVI video file.
- $F \leftarrow$ Decoding & filtering frames (Y).
- $N \leftarrow$ Number of frames (F).

- For $j = 1$ to N
- $B \leftarrow$ Dividing frame into 32×32 ($F[j]$)
- $D \leftarrow$ Calculate standard deviation for all blocks(B)
- $A \leftarrow$ Finding average standard deviation (D)
- $L \leftarrow$ Get blocks with standard deviation less than average (B,D,A)
- $M \leftarrow$ Creating map block (L)
- $Z \leftarrow$ Embedding (X) into (L) by LSB.
- End For
- Return (Y).

3.8 Secret and cover video file identification:

When the user wants to send secret text to other receiving party, he/she runs the program. After the program starts the user writes his/her secret message , after that the user clicks on “select a video” button to choose a cover file, a dialog will open allowing the user to select any AVI video file only.

Create stego video file:

Hiding the secret text message inside the cover video file occurs in this step,

the hiding process is done by using the Least Significant Bit (LSB) steganography method, LSB method works by storing all the bits of the secret text into the right most bit in the cover video file.

This step includes LSB hiding algorithm.

LSB Hiding Algorithm

The LSB algorithm hides the bits of the secret text into the cover video file bits, The first step is getting the secret text message and appending a special escape sequence to it (\$\$##@ @^^), the escape sequence is essential to identify the end of the secret text in the retrieving process.

After preparing the secret text, the cover video file is decoded into frames, each frame will be divided into 32×32 pixels blocks, then the standard deviation will be calculated for each block. After that, the block with standard deviation greater than the average will be excluded, after preparing the blocks in the frame, the secret text bits will be substituted in these blocks.

The process is shown in the Figure (3-3).

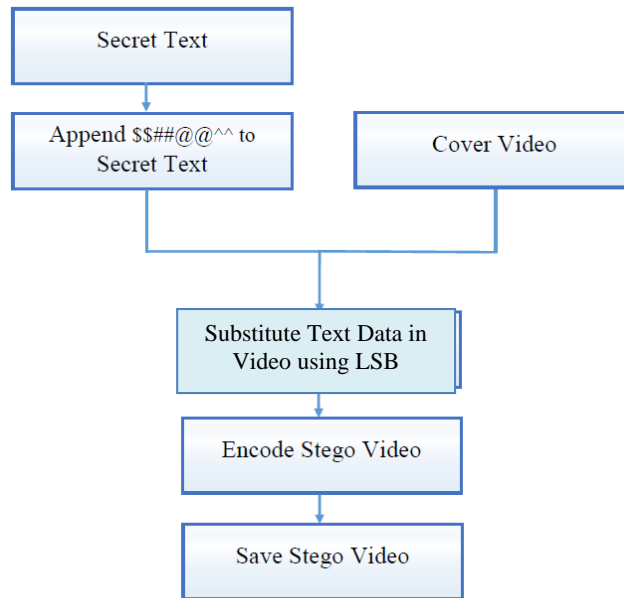


Figure (3-3) ... Hide-LSB

Algorithm: LSB Hiding algorithm

// Input algorithm: Modified secret text and frames blocks.

// Output algorithm: Stego video file.

- $X \leftarrow$ blocks from the cover video file.
- $Y \leftarrow$ Modified secret text.
- $L \leftarrow$ Length of (Y)
- For $i = 1$ to L
- $Z \leftarrow$ binary (Y[i])
- For $j = 1$ to 8

- $\text{LSB}(X) \leftarrow Z[j]$
- end for
- end for
- return (X)

Save Stego Video

After substituting secret text bits in the blocks that has standard deviation less than or equal the average, the frames will be encoded into a raw video, raw video doesn't manipulate pixels value therefore the substituted data will not be affected by encoding.

The resultant stego video file contains the secret text hidden in it, this stego video can be sent safely via any medium to receiving participant, and the receiving party then can extract the substituted secret text from the video.

3.9 Receiver procedure

The receiving procedure contains all the steps that occurs at the receiver side, this step starts from decoding the raw stego video and ends with retrieving the hidden secret text within the stego video.

The receiver procedure includes the following stages:

1. Extracting hidden text from the raw stego video file.
2. Showing the extracted secret text.

Receiver Algorithm:

The receiver algorithm involves extracting secret text from the raw stego video file as shown in figure (3-4):-

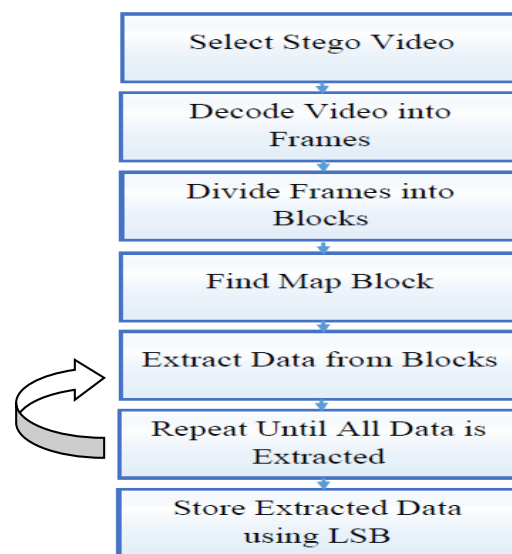


Figure (3-4) ... Receiver Procedure

Algorithm: Receiver algorithm:-

// Input algorithm: - Stego raw video file.

// Output algorithm: - Extracted secret text.

- $Z \leftarrow$ Stego raw video file.
- $F \leftarrow$ Decoding frames (Z).

- $N \leftarrow$ Number of frames (F).
- For $j = 1$ to N
- $B \leftarrow$ Dividing frame into 32×32 (F[j])
- $D \leftarrow$ Calculate standard deviation for all blocks(B)
- $A \leftarrow$ Finding average standard deviation (D)
- $L \leftarrow$ Get blocks with standard deviation less than average (B,D,A)
- $M \leftarrow$ Read map block (L)
- $S \leftarrow$ Exclude map block and get all other frame blocks(D,M)
- $K \leftarrow$ Number of blocks (S).
- For $i=1$ to K
- If (Block Contains Data (M,S[i]))
- $Z \leftarrow Z +$ (Extract text from (S[i]) by LSB).
- End If
- End For //i
- End For //j
- Return (Z).

Extract stego video

When the receiving party chooses to retrieve a stego raw video file, the retrieving procedure extracts the hidden secret text from the video, this procedure starts by extracting secret text bits from the blocks of the decoded frames of the video,

and the retrieving procedure retrieves the least significant bit from the raw stego video file and forms the secret text.

This step includes LSB retrieving algorithm.

LSB Extraction Algorithm

This algorithm works on the stego raw video file that has secret text hidden within it, the algorithm works on the frame divided blocks and forms the secret text out of the least significant bits of those blocks, this algorithm keeps extracting bits until it reads the following sequence of characters “\$\$##@@^^”, after reading them the algorithm stops reading blocks and outputs the secret text into the text area, the procedure steps is shown in the Figure 3-5.

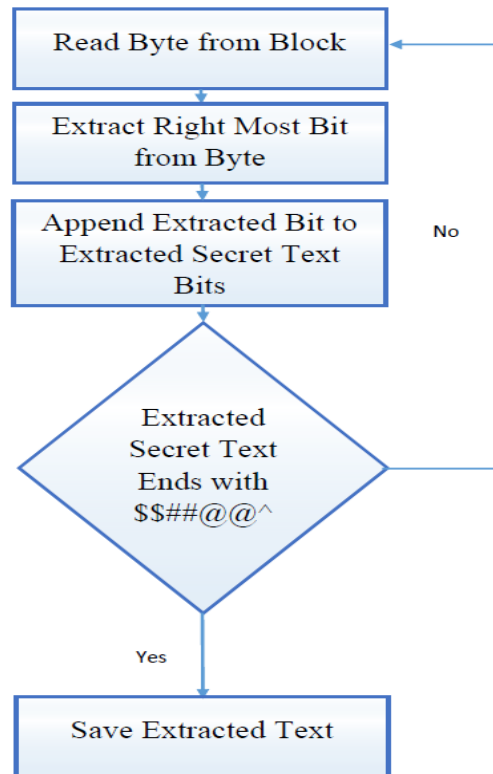


Figure (3-5) ... LSB Extraction

Algorithm: LSB Extraction algorithm

// Input algorithm: Frames blocks of stego raw video file.

// Output algorithm: Secret text file.

- $X \leftarrow$ blocks from the stego raw video file.
- $N \leftarrow$ Number of blocks(X)
- For $i=1$ to N

- $Y \leftarrow Y + \text{LSB}(X[i])$
- If Y ends with “\$\$##@@^^”
- Exit for
- End if
- end for
- return (Y)

Retrieve secret text

This application uses the previous algorithm for extracting text from the stego video and shows it in the text area. This secret text file is hidden by the sender side.

It is preferable that the receiving party does not save a copy of the saved secret text and depend on extracting the secret text from the video when needed, this approach increases the security of our model, and the secret text should be saved in receiver device as stego not as original form.

Limitation

The application was implemented using Microsoft .Net C#, so it works on any machine running Microsoft Windows,

this allows sender to hide text in video file and send it to any recipient that uses Microsoft Windows, and the stego video file could be transferred to receiver by any means, internet, flash memory, DVDs.

The cover video file should be an AVI file and the output stego video is AVI raw video, the raw video file takes a lot of space to store because of the lossless encoding, other video types makes modifications of the frames bytes while encoding thus any bits stored in the frames will be subject to altering and this operation is not reversible, which mean that once the video is decoded the decoding procedure modifies frame bits and the secret bits are lost.

Chapter four

The Experimental Works

Chapter four

The Experimental Work

This chapter introduces a model for transmitting secret text between different parties; this model presents a procedure to prevent unwanted parties from reading secretly transmitted text over a network between any two devices that uses Microsoft Windows.

4.1 Preface:

The application that implements this model was tested on two separate machines running Microsoft Windows. Those machines are connected to the internet. The application starts at the sender side, by hiding text inside a cover video, then the application creates a new video (stego video file) that can be safely transmitted via the internet to any other party.

Based on that, the proposed model divided in to two parts: the first part is called Sender, which takes place at the sender side, whom has secret text to send to the receiving party. And the second part is called Receiver which takes place at the receiver side that receives a stego video and retrieves the hidden text from it. As shown in Figurer (4-1).

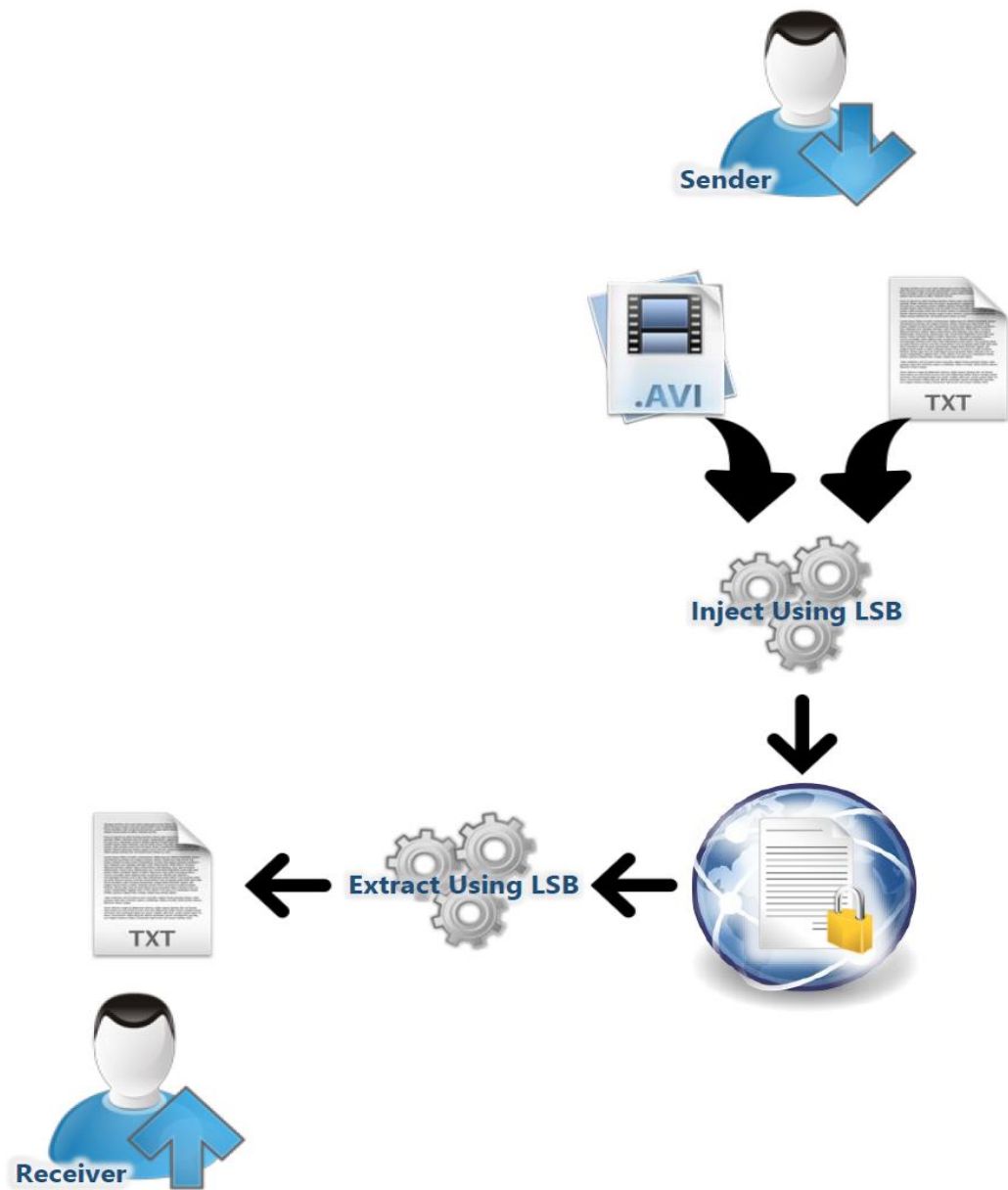


Figure (4-1) ... Steps of Model

4.2 Sender part:

The sender uses the application on Microsoft Windows OS, and runs it, this application is called “FirstParty”, the application UI is shown in Figure (4-2):-

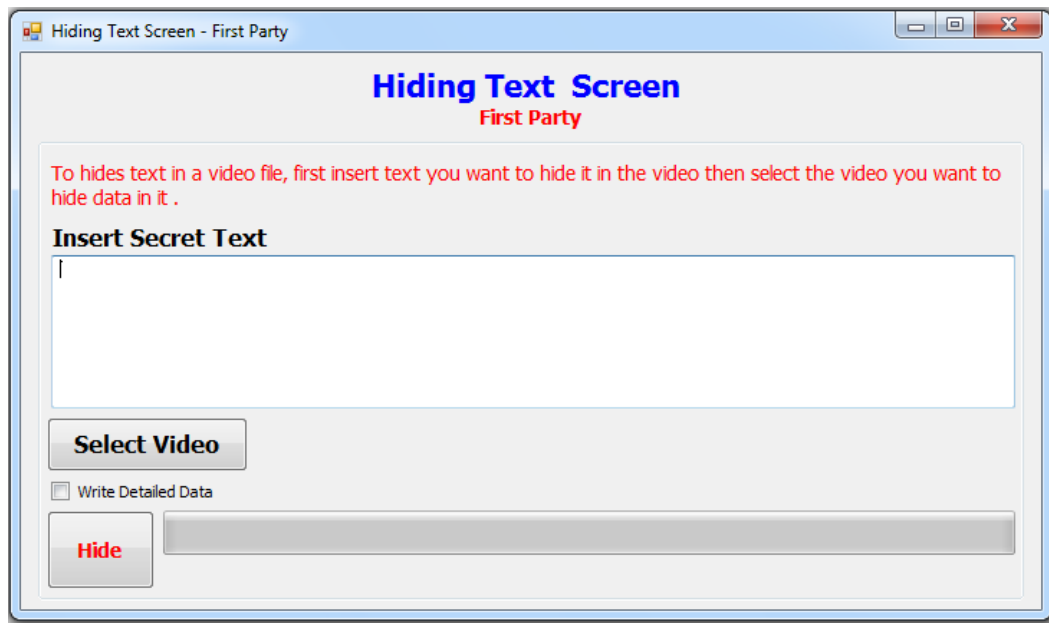


Figure (4-2) ... Application Interface

The applications has hiding mode, the sender should use this mode which contains the following functions: secret text, Select video and (Write detailed data is intended to be used for illustration purposes).

When the user wants to send a secret text he/she do the following:

The sender prepares a cover video file, this video file should be an AVI file,

the cover file will be used later to substitute the secret text in it with the LSB.

The user starts the application from a machine running Microsoft Windows OS.

The cover video is selected by browsing AVI files from the “Select Video” function as shown in Figure (4-3).

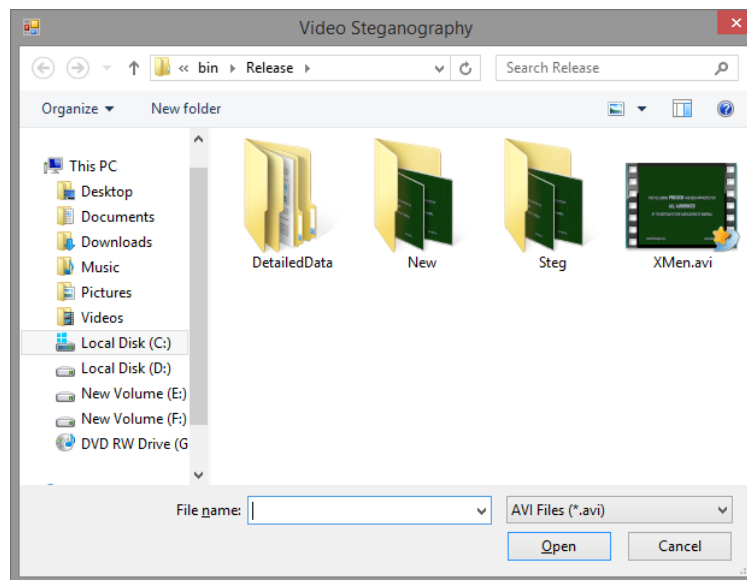


Figure (4-3) ... Select Cover Video

The secret text is written in the secret text area as shown in Figure (4-4):

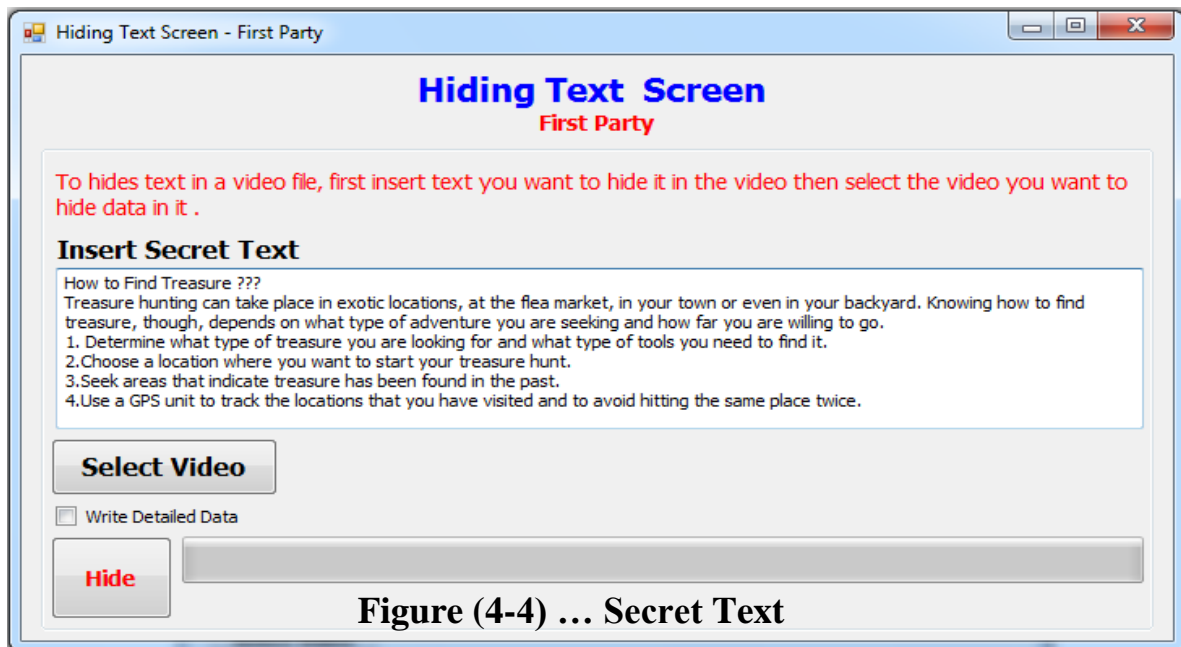


Figure (4-4) ... Secret Text

The user starts the hiding function to produce AVI stego file by clicking on “Hide”, the progress is shown on the UI as shown in Figure (4-5).

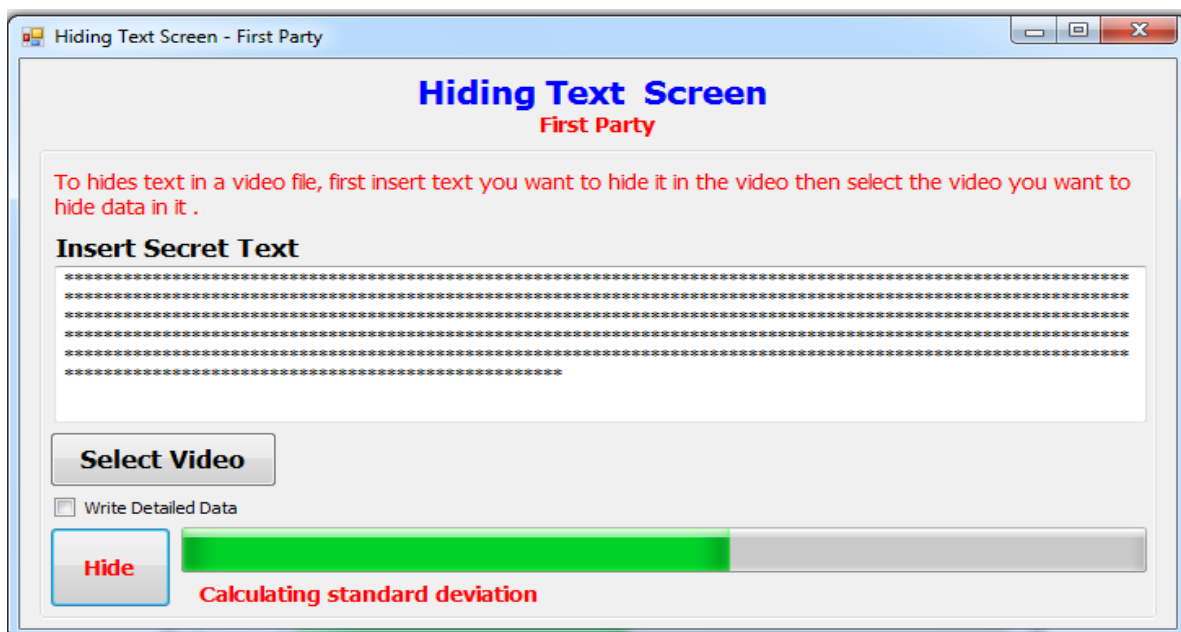


Figure (4-5) ... Hidden Progress

The letters of secret message will convert to stars letter, after clicking on "hide" button to prevent anybody to see the secret message. The application substitutes the secret text with the LSB in the cover video file to produce the stego video file. The user can now transfer the file to the receiving party by any means; he/she just needs to send the output file.

The cover file and the stego file does not have any visual difference because of substituting text with the LSB within the areas that will not have any visual impact, the frames are visually identical and the differences cannot be noticed as shown in Figure (4-6).

This is the original frame:



Figure (4-6 A)

This is the frame after substituting the secret text in it:



Figure (4-6 B)

Note: we can find the original images in directory name "New" , where we can find the images after substituting the secret text with the LSB in directory name "Steg" .

The histogram shows the distribution of (RGB) light and dark pixels in the image, in the Figure (4-7) shows the histograms for two images before and after hiding the text:

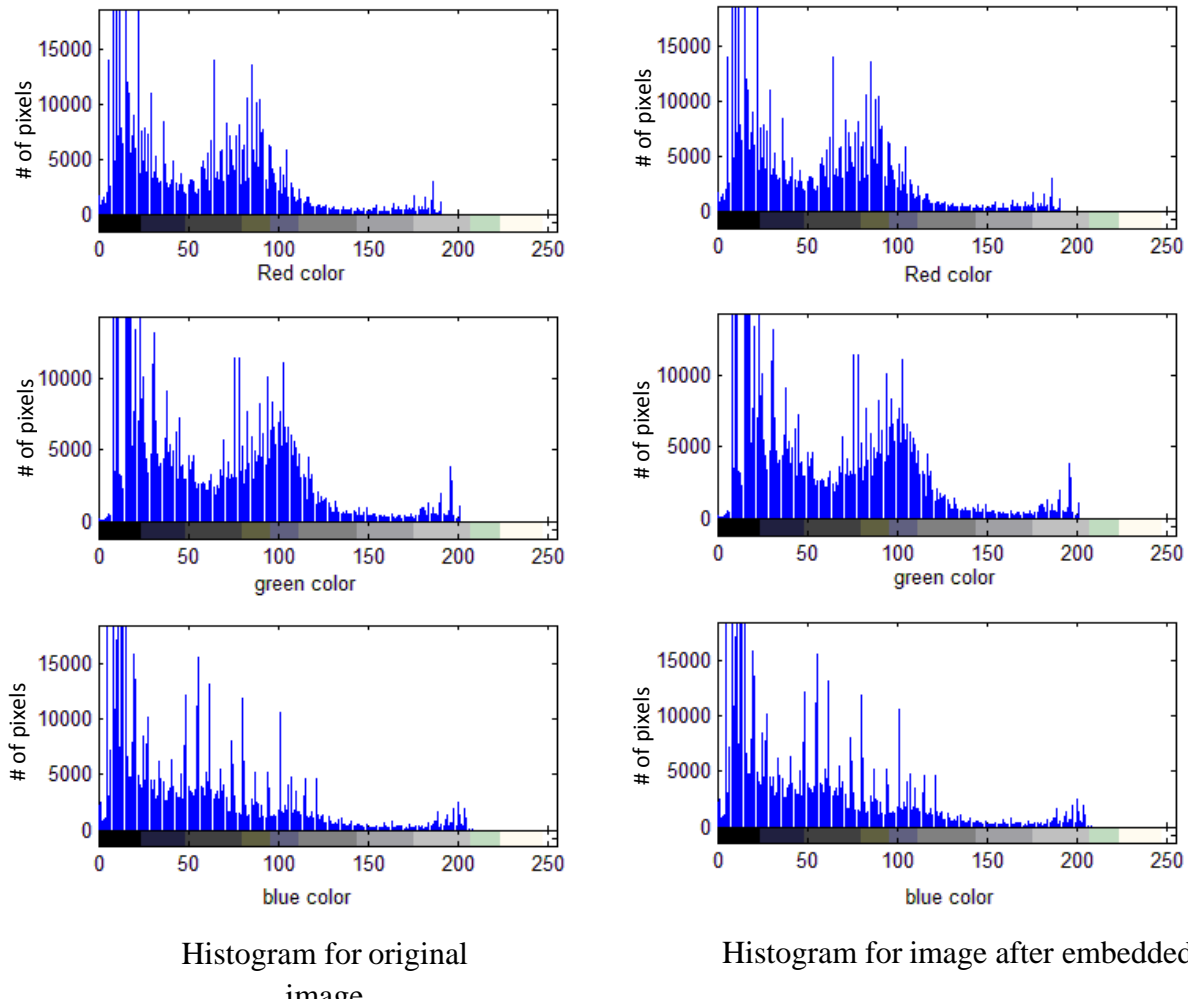


Figure (4-7) ... distribution of (RGB) color

We note in Figure (4-7), there is no difference between two histograms, this leads to the difficulty of discovering the hidden texts , and this will prove the efficiency of the algorithm used in the process of concealment .

Figure (4-8) shows the blocks that the application will use to substitute the secret data with the LSB in it, the blocks are located where there are not much details and where there are shadows and dark areas, altering the LSB in these areas will not affect the resultant frame as shown before.

Figure (4-8) shows the blocks in which the data is substituted with the LSB, areas in white are the blocks that contains data in its LSB, you can notice that the blocks are located where there are no much details and where the shadows are.



Figure (4-8) ... Used Blocks

The secret text will not hide in consecutive frames to be more secure, the program will use the frames # as the following:

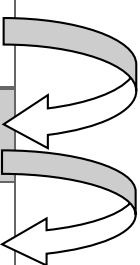
Frames # : 1 , 4 , 7 , 10 , 13 , 16 , 19 , 22

Example:

In this example the following text will be embedded in a block, the secret text is “secret” without the quotes.

First the text “secret” will be converted to ASCII code so that it can be converted to binary, in the following is the conversion table:

Letter	s	e	c	r	e	t
ASCII	115	101	99	114	101	116
Binary	01110011	01100101	01100011	01110010	01100101	01110100



The binary value for each letter will be substituted with the LSB in the image block.

After that a frame will be extracted from the video, and will be divided into blocks as shown in Figure (4-9).

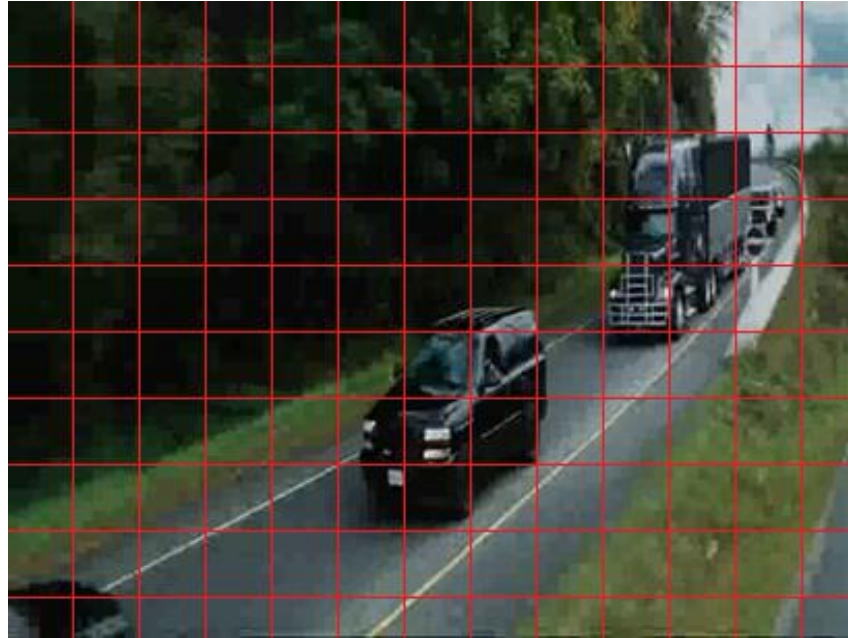


Figure (4-9) ...Frame Blocks Division

Each block consists of 32×32 pixels, and then the standard deviation will be calculated for all the blocks as shown in the following example:

We will calculate the standard deviation for the first block as shown in Figure (4-10):

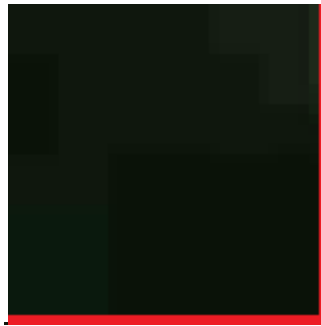


Figure (4-10) ...Frame Block

The first pixel RGB value is (15, 23, 13) and the second is (10, 18, 8) and so on for the rest of the 1024 pixels (32×32).

The standard deviation is calculated as following:

σ = Standard deviation

$$\begin{aligned}\sigma &= \sqrt{\text{Variance}} = \sqrt{\sigma^2} \\ &= \sqrt{\sum_{i=1}^n (k_i - \hat{k})^2 P_i}.\end{aligned}$$

Where the variance is the average of the squared differences from the Mean.

For the previous example:

$$\text{Mean} = ((15+23+13) + (10+18+8) + \dots) / 1024$$

$$\text{Variance} = ((15+23+13) - \text{Mean})^2 + ((10+18+8) - \text{Mean})^2 + \dots$$

$$\text{Standard Deviation} = \sqrt{\text{Variance}}$$

After calculating the standard deviation for all the blocks we will exclude the blocks with standard deviation greater than the average and stand substituting data in the remaining blocks.

Substituting the binary representation of the secret text “secret” is done as following:

The following block has a standard deviation less than or equal to the average as it appear in Figure (4-11)

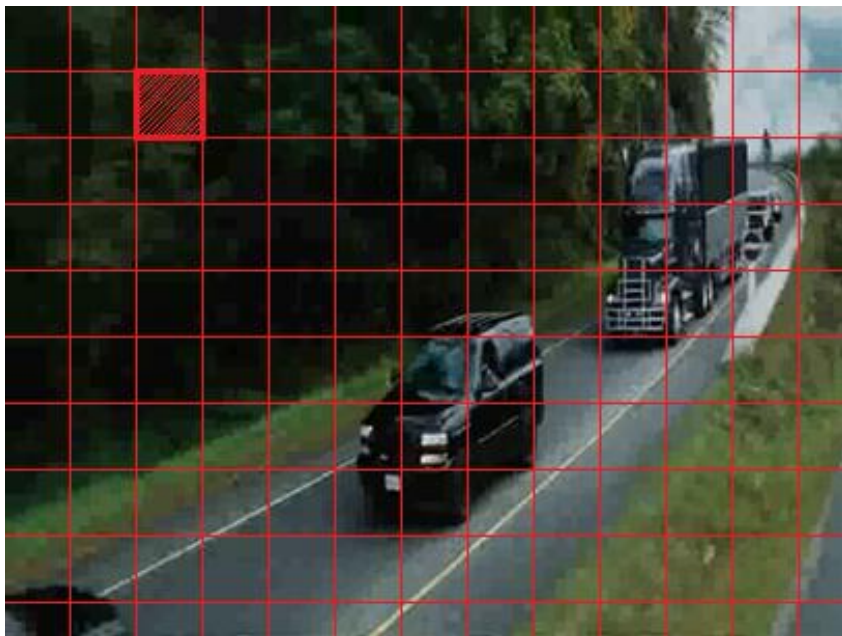


Figure (4-11) ...Chosen Frame Block

The pixel values are: (34,42,28), (35,43,29),(43,52,35),...

The binary representation for them is: (00100010,00101010,00011100),
 (00100011,00101011, 00011101), (00101011, 00110100, 00100011) ...

The LSB is highlighted in bold, which the secret text will be substituted with it.

Substituting the first letter “s” from the secret text “secret” is done as follow:

s = 01110011 in binary, the first bit (1) will be substituted with the first pixel LSB (34), after substituting the first pixel value will be 34 (00100010) and the second value will be 42 (00101011) and so on, the pixels values for the above block will be:

(00100010,00101011,00011101) , (00100011,00101010,00011100),
(00101011, 00110101, 00100010) ...

The result of pixels values are: (34,43,29), (35,42,28), (43, 53,34) ...

4.3 Receiver part:

This part is used by the receiver to extract data that was previously substituted with the LSB in a video file. When the user wants to retrieve a secret text that was hidden in a stego video file he/she do the following:

The receiver saves the stego video; this file will be used later by the application to extract the data from it.

The user starts the application which is called “Second Party” , from a machine running Microsoft Windows OS and selects the stego video by browsing AVI files from the “Select Video” function as shown in Figure (4-12).

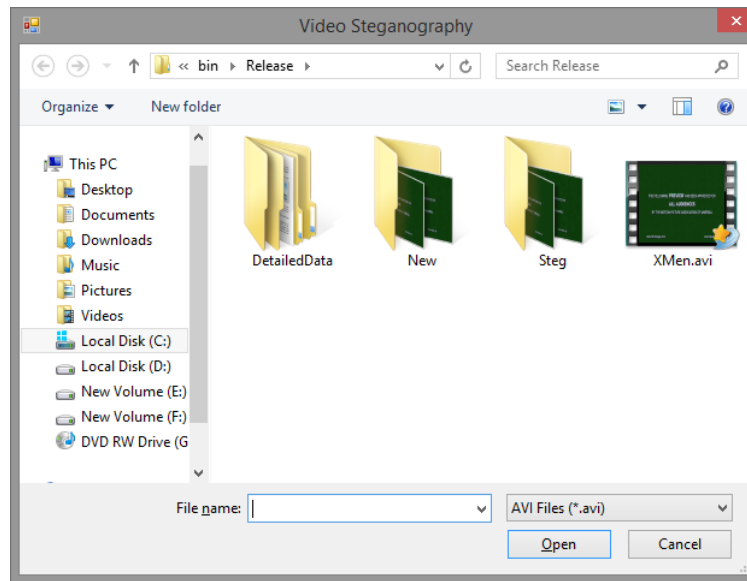


Figure (4-12) ... Select Stego Video

The user starts the retrieving function to extract the secret text by clicking on “Retrieving”

The progress is shown on the UI as shown in Figure (4-13)

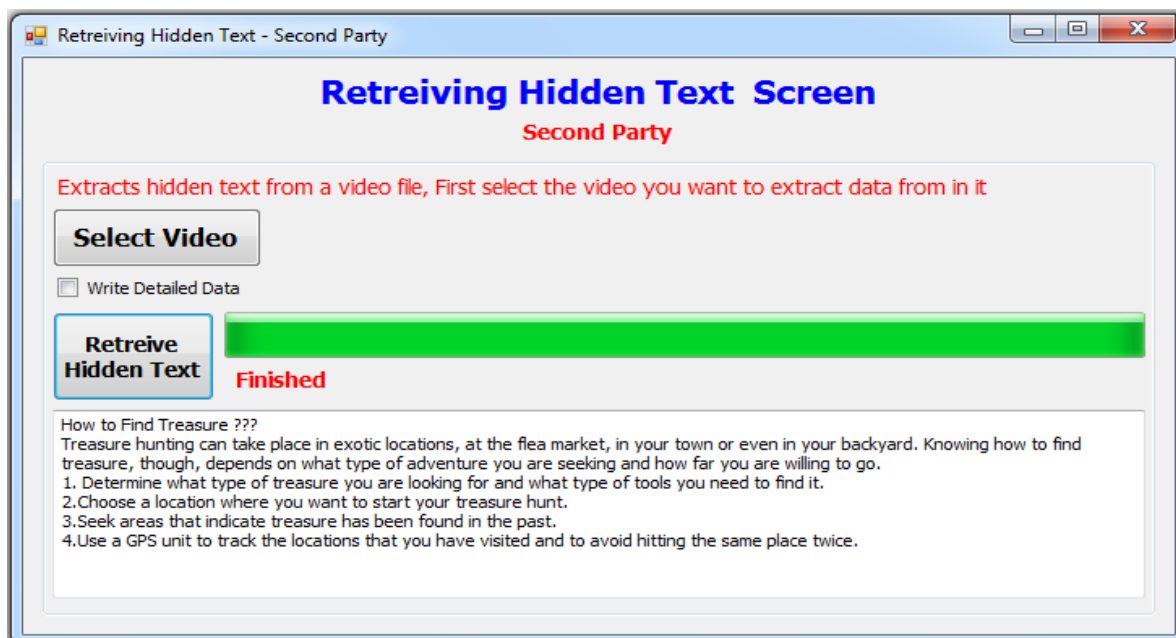


Figure (4-13) ... Extraction Progress

After finishing, the application shows the extracted text in a text area, this text is identical to the secret text file created by the sender.

Example :

In Receiver part, After calculating the standard deviation for all the blocks, will exclude the blocks with standard deviation greater than the average , and makes the processes on remain blocks to retrieve the secret message .

The pixels values : (34,43,29), (35,42,28), (43, 53,34) ... are after embedded the letter "s" in above example , the receiver part will make the following steps to retrieve the secret message :

1. convert these pixels values to binary as following :

(00100010,00101011,00011101) , (00100011,00101010,00011100),
 (00101011, 00110101, 00100010)

2. Retrieve The LSB is highlighted in bold, which the secret text will be substituted in as following :

(0010001**0**,0010101**1**,0001110**1**) , (0010001**1**,0010101**0**,0001110**0**),
 (0010101**1**, 0011010**1**, 00100010) ...

0 1 1 1 0 0 1 1

3. (01110011) in binary , then Converted it to ASCII code so that it can be converted to the letter as shown below :

Binary	01110011	
ASCII	115	
Letter	s	

Substituting secret text in the cover video file includes: decoding video, dividing frames, calculating standard deviation, creating map block, converting secret text to binary, substituting data with the LSB in cover video and encoding frames to create stego video.

The extraction process includes: decoding stego vide, dividing frames to blocks, calculating standard deviation, read map block, extracting data from LSB and storing extracted data.

Hiding Detailed Example:

Following is a detailed example of hiding some secret text into frame:

1. When hiding text make sure to check the “Write Detailed Data” checkbox as in Figure (4-14).

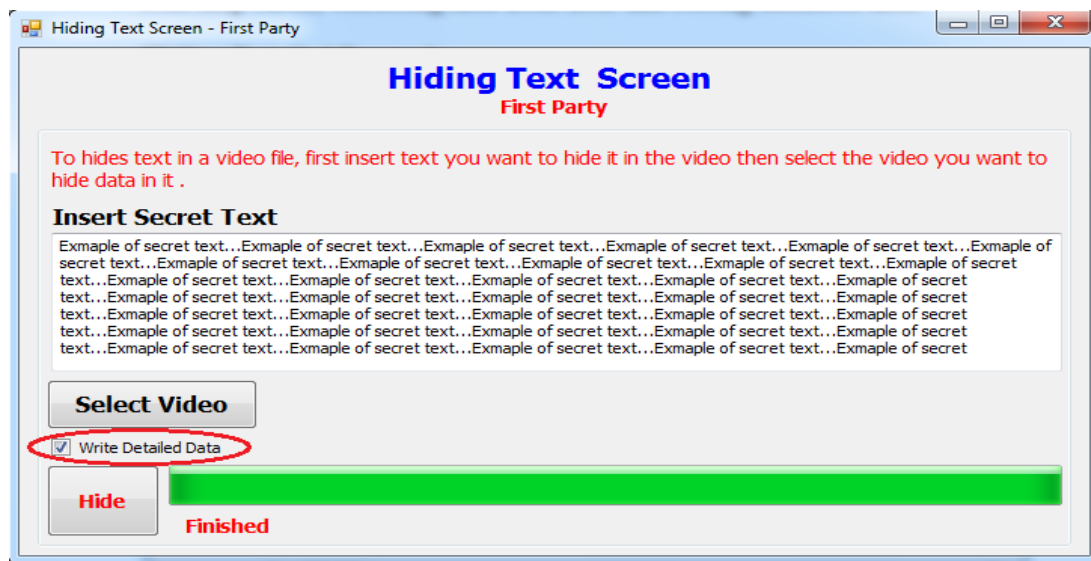


Figure (4-14) ... Write Detailed Data

2. After hiding secret text a special directory will be created that holds the detailed hiding text process, this directory name is “Detailed Data” in which you can find "Hidden" directory.

3. In the hiding secret text process the “Hidden” directory will contain detailed analysis data for each frame, in which you can see where the map block is located in that frame, the blocks that were used to hide the text and the hidden text in each frame.

Following is an example:

Blocks: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, ...

Block map: 41

0:1 1:0 2:0 3:0 4:0 5:0 6:0 7:0 8:1 9:0 10:0 11:1 12:0 13:1 14:1 15:1 16:0
 17:0 18:0 19:1 20:1 21:0 22:1 23:0 24:0 25:0 26:1 27:0 28:1 29:1 30:1

Blocks With Data: 0,8,11,13,14,15,19,20,22,26,28,29,30,33,34,36,37,34,44,45,48,49...

0 Example of secret text...Example of secret text...Example of secret text...Example of
 8 f secret text...Example of secret text...Example of secret text...Example of secret te
 11 text...Example of secret text...Example of secret text...Example of secret text...Exam
 13 ample of secret text...Example of secret text...Example of secret text...Example of se
 14 cret text...Example of secret text...Example of secret text...Example of secret text
 15 xt...Example of secret text...Example of secret text...Example of secret text...Exampl
 19 ple of secret text...Example of secret text...Example of secret text...Example of secr
 20 cret text...Example of secret text...Example of secret text...Example of secret text..
 22 ...Example of secret text...Example of secret text...Example of secret text...Example
 26 e of secret text...Example of secret text...Example of secret text...Example of secret
 28 et text...Example of secret text...Example of secret text...Example of secret text...E

29 .Example of secret text...Example of secret text...Example of secret text...Example of
30 of secret text...Example of secret text...Example of secret text...Example of secret t
33 text...Example of secret text...Example of secret text...Example of secret text...Exa
34 xample of secret text...Example of secret text...Example of secret text...Example of s
36 secret text...Example of secret text...Example of secret text...Example of secret tex
37 ext...Example of secret text...Example of secret text...Example of secret text...Examp
43 mple of secret text...Example of secret text...Example of secret text...Example of sec
44 cret text...Example of secret text...Example of secret text...Example of secret text.
45 t...Example of secret text...Example of secret text...Example of secret text...Example
48 le of secret text...Example of secret text...Example of secret text...Example of secre
49 ret text...Example of secret text...Example of secret text...Example of secret text...

The blocks part shows how many blocks are there in the image.

The block map shows where the map block is located in the previous example the "map block" is the block number "41".

After the blocks map you can find a list of all the block and an indication if the block was used to hide text in it or not ,

resembled in 0 or 1 where (0) indicated that the block was not used to hide secret text in it, and (1) indicated that the block was used to hide secret text in it , in the previous example we can see that the block number 0 is used but the blocks 1,2,3,4,5,6,7 are not used where the block 8 is used and so on.

After that you can see a list of each of the used blocks and the secret text that were hidden within it.

4. In the extraction process the same detailed data is followed except the frames with hidden text will have a detailed extraction file.

Chapter five

Conclusion

Chapter five

Conclusion

This chapter provides the final conclusions of the research, plus the suggested future works.

5-1 Conclusion:

The hiding process was applied to several different videos of AVI type specifications (Frame width=640, Frame height=480) , using (Microsoft.Net C#), and the results were as following:

1. Results proved the efficiency of the algorithm, where the hidden information did not cause any distortion on the video cover used , and no difference between histograms for two images before and after embedded data.
2. Cover image of the division led to a series of passages to increase the storage space possible to store confidential texts.
3. Hiding in the standard deviation less of average sections is better than larger values, because the less standard deviation has a less dispersion of data.

4. The retrieval of the secret message was full without lossing of any hidden data.
5. The hiding process took long time.

5-2 Future works:

It is possible to enhance the hiding procedure by using the following suggestions:

1. Using an innovative way to compress the video output to return it to the original size without losing some of the data.
2. Enhance the procedure by hiding a water mark to prove authentication.
3. Using different methods of steganography to improve security and reduce size.

References:

1. Alalem, M., & Manasrah, A. (2008). *A Steganographic Data Security Algorithm with Reduced Steganalysis Threat*, Birzeit University, Birzeit, Palestine.
2. Al-Neamah, R., & Al-Neamah, S. (2010). Design and Implementation of Steganographic Algorithm on Video File (mov), *Al-Rafideen Computer Science and Mathematics Journal*, November, 227-237.
3. Beenish, M., & Faruqui, R. (2008). *A Steganography Implementation*, 11th conference of TENCON, IEEE Region, Taipei.

4. Chang, W. & Tai, W. (2012). Histogram-based Reversible Data Hiding Based on Pixel Differences with Prediction and Sorting, *KSIIT Transactions on Internet and Information Systems*, 6 (12): 3100-3116.
5. Chen, M., Zhang, R., Xinxin, N., & Yang, X. (2006). *Analysis of Current Steganography Tools: Classifications & Features*, International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), USA.
6. Chen, P., & Lin, H. (2006). A DWT Based Approach for Image Steganography, *International Journal of Applied Science and Engineering*, 4 (3): 275-290.
7. Cole, E. (2003). *Hiding In Plain Sight: Steganography and the Art of Covert Communication*, Indianapolis, Indiana: Wiley Publishing, Inc., USA.
8. Dennis, A., Wixom, B., & Tegarden, D. (2004). *Systems Analysis and Design with UML*, Version 2.0.
9. Deshpand, N. & KamalapurSnehal, J. (2004). *Implementation of LSB Steganography and Its Evaluation for Various Bits*, Digital Information Management, 1st International Conference, Bangalore.
10. Jalab, H., Zaidan, A., & Zaidan, B. (2009). Frame Selected Approach for Hiding Data within MPEG Video Using Bit Plane Complexity Segmentation, *Journal of Computing*, 1 (1): 108-113.

11. Jian, A., Gupta, I. (2007). *A JPEG Compression Resistant Steganography Scheme for Raster Graphics Images*, 10th conference of TENCON, IEEE Region, Taipei.
12. Johnson, N., & Jajodia, S. (1998). *Steganalysis of Images Created Using Current Steganography Software*, Center for Secure Information Systems, George Mason University, Fairfax, Virginia, USA.
13. Juneja, M., & Sandhu, P. (2009). *Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption*, International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, Kerala.
14. Khan, W., Kumar, S., Gupta, N., & Khan, N. (2011). A Proposed Method for Image Retrieval using Histogram values and Texture Descriptor Analysis, *International Journal of Soft Computing and Engineering (IJSCE)*, 1 (2): 33-36.
15. Krenn, J. (2004). *Steganography and Steganalysis*, Tata Institute of Fundamental Research, India.
16. Kumar, W., & Mahesh, P. (2007). *Security through obscurity: Steganography*, Department of Computer Science & Systems Engineering, Andhra University, India.

17. Kurah, C. & Mchughes, J. (1992). *A cautionary note on image downgrading*, In Proceedings of the IEEE Computer Security Applications Conference, Vol. 2., IEEE Computer Society Press, Los Alamitos, CA, 153–159.
18. Lee, G., Yoon, E., & Yoo, K. (2008). *A new LSB based Digital Watermarking Scheme with Random Mapping Function*, in 2008 IEEE DOI 10.1109/UMC.33.
19. Mathkour, H., Al-Sadoon, B., & Tourir, A. (2008). *A New Image Steganography Technique*, 4th International Conference of Wireless Communications, Networking and Mobile Computing, WiCOM, Dalian.
20. Morkel, T., Eloff, J., & Oliver, M. (2005). *An overview of image steganography*, proceeding of fifth annual information security South Africa conference, ISSA, 1-11.
21. Paar, C. (2005). *Applied cryptography and data security*, version 2.5, Ruhr University at Bochum, Germany.
22. Po-Yueh, C., & Hung-Ju, L. (2006). *A DWT Based Approach for Image Steganography*, *International Journal of Applied Science and Engineering*, 4 (3): 275-290.
23. Provos, N., & Honeyman, P. (2003). *Hide and Seek: An Introduction to Steganography*, University of Michigan, USA.

24. Raftari, N., & Moghadam, A. (2012). *Digital Image Steganography Based on Assignment Algorithm and Combination of DCT-IWT*, 4th International Conference on Computational Intelligence, Communication Systems and Networks, Phuket.
25. Raja, K., Chowdary, C., Venugopal, K., Patnaik, L. (2005). *A Secure Image Steganography using LSB, DCT and Compression Techniques on Raw Images*, 3rd Conference of Intelligent Sensing and Information Processing, 14-17 December, India.
26. Ramalingam, M. (2011). Stego Machine – Video Steganography using Modified LSB Algorithm, *World Academy of Science, Engineering and Technology*, 50, 497-500.
27. Sharma, V., & Shrivastava, V. (2012). A Steganography Algorithm for Hiding Images by improved LSB substitution by minimize detection, *Journal of Theoretical and Applied Information Technology*, 36 (1): 1-8.
28. Thota, N., & Devireddy, S. (2008). Image Compression Using Discrete Cosine Transform, *Georgian Electronic Scientific Journal: Computer Science and Telecommunications*, 3 (17): 35-43.
29. Tirkel, A., Rankin, G., Schyndel, R., Ho, W., Mee, N., & Osborne, C. (1993). *Electronic watermark*, In Proceedings of DICTA, 666–672.

30. Titty, T. (2009). *Steganography: Reversible Data Hiding Methods for Digital Media*, Bachelor project, USC University, USA.
31. Walia, E., & Navdeep, J. (2010). An Analysis of LSB & DCT based Steganography, *Global Journal of Computer science & technology*, Vol. 10, Issue 1 (Ver 1.0), April.
32. Wang, H., & Wang, S. (2004). *Cyber Warfare: Steganography vs. Steganalysis*, Vol 47, Communications of the ACM, October.
33. Zheng, D., Liu, Y., Zhao, J., & El Saddik, A. (2007). *A Survey of RST Invariant Image Watermarking Algorithms*, ACM.