Independent Domain of Symmetric Encryption using Least Significant Bit

Bhupathi Rajan

2011
## DEGREE PROJECT
Computer Engineering

<table>
<thead>
<tr>
<th>Programme</th>
<th>Reg number</th>
<th>Extent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Masters Programme in Computer Engineering - Applied Artificial Intelligence</td>
<td>E4037D</td>
<td>15 ECTS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name of student</th>
<th>Year-Month-Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bhupathi Rajan</td>
<td>2011-10-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supervisor</th>
<th>Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diala Jomma</td>
<td>Hasan Fleyeh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Company/Department</th>
<th>Supervisor at the Company/Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department of Computer Engineering, Dalarna University</td>
<td>Diala Jomma</td>
</tr>
</tbody>
</table>

### Title
Independent Domain of Symmetric Encryption using Least Significant Bit

### Keywords
- Image Watermarking
- Canny Edge Detection
- LSB (Least Significant Bit)
- Vigenere Square
- PSNR (Peak Signal to Noise Ratio)
Abstract

The rapid development of data transfer through internet made it easier to send the data accurate and faster to the destination. There are many transmission media to transfer the data to destination like e-mails; at the same time it is may be easier to modify and misuse the valuable information through hacking. So, in order to transfer the data securely to the destination without any modifications, there are many approaches like cryptography and steganography. This paper deals with the image steganography as well as with the different security issues, general overview of cryptography, steganography and digital watermarking approaches.

The problem of copyright violation of multimedia data has increased due to the enormous growth of computer networks that provides fast and error free transmission of any unauthorized duplicate and possibly manipulated copy of multimedia information. In order to be effective for copyright protection, digital watermark must be robust which are difficult to remove from the object in which they are embedded despite a variety of possible attacks.

The message to be send safe and secure, we use watermarking. We use invisible watermarking to embed the message using LSB (Least Significant Bit) steganographic technique. The standard LSB technique embed the message in every pixel, but my contribution for this proposed watermarking, works with the hint for embedding the message only on the image edges alone. If the hacker knows that the system uses LSB technique also, it cannot decrypt correct message. To make my system robust and secure, we added cryptography algorithm as Vigenere square. Whereas the message is transmitted in cipher text and its added advantage to the proposed system. The standard Vigenere square algorithm works with either lower case or upper case. The proposed cryptography algorithm is Vigenere square with extension of numbers also. We can keep the crypto key with combination of characters and numbers. So by using these modifications and updating in this existing algorithm and combination of cryptography and steganography method we develop a secure and strong watermarking method.

Performance of this watermarking scheme has been analyzed by evaluating the robustness of the algorithm with PSNR (Peak Signal to Noise Ratio) and MSE (Mean Square Error) against the quality of the image for large amount of data. While coming to see results of the proposed encryption, higher value of 89dB of PSNR with small value of MSE is 0.0017. Then it seems the proposed watermarking system is secure and robust for hiding secure information in any digital system, because this system collect the properties of both steganography and cryptography sciences.
ACKNOWLEDGMENT

I begin with conveying my deepest gratitude to the Swedish government, for giving me an opportunity to pursue a Master’s degree in Computer Science.

I would like to thank my supervisor Mrs. Diala Jomaa for their continuous support and guidance for my thesis work. She is very kind to me and approachable for any doubts and suggestions. Without her valuable suggestions, I guess this thesis would not be possible. I sincerely thank her, for their endless source of ideas and valuable hours of guidance for the past ten months.

I would like to thank Mr. Hasan Fleyeh, Mr. Siril Yella, Mr. Pascal Rebreyend, and Mr. Mark Dougherty for their encouragements and valuable teachings. Finally I would like to convey my deepest love to my parents, friends for their blessings and wishes, and lovable kids who made me to smile always.
# TABLE OF CONTENTS

## Chapter 1

**INTRODUCTION** .......................................................................................................................... 1

1.1 Motivation ................................................................................................................................. 1
1.2 Intellectual Property and the Digital Age ............................................................................... 1
1.3 Problem Statement .................................................................................................................. 2
1.4 Objective .................................................................................................................................. 3
1.5 Proposed Solution to Prevent the Security Attacks ......................................................... 4
1.6 Content of the Project .......................................................................................................... 5

## Chapter 2

**THEORETICAL BACKGROUND** ......................................................................................... 6

2.1 Literature Review ................................................................................................................... 6
2.2 Data hiding techniques .......................................................................................................... 7
2.3 Cryptography ......................................................................................................................... 8
2.4 Cryptography Algorithms ..................................................................................................... 9
   2.4.1 Symmetric Encryption ................................................................................................. 9
   2.4.2 Asymmetric Encryption ............................................................................................. 11
   2.4.3 Vigenere Square ......................................................................................................... 12
      2.4.3.1 Encoding ............................................................................................................ 12
      2.4.3.2 Decoding ............................................................................................................ 13
2.5 Steganography ...................................................................................................................... 14
2.6 LSB Technique .................................................................................................................... 16
2.7 Digital Watermarking ........................................................................................................... 16
2.8 Computing the PSNR and MSE ........................................................................................... 18

## Chapter 3

**PROPOSED WATERMARKING SCHEME** ........................................................................... 19

3.1 Developed Watermarking Scheme ....................................................................................... 19
3.2 Cryptographic Algorithm ..................................................................................................... 22
   3.2.1 Extension of Vigenere Square ................................................................................... 22
3.3 Steganography Method ......................................................................................................... 23
   3.3.1 Canny Edge Detection ............................................................................................... 23
   3.3.2 LSB Substitution Techniques ..................................................................................... 24
3.4 LSB based Steganography .................................................................................................... 25
   3.4.1 Algorithm to embed the Watermark as text message ................................................... 25
   3.4.2 Algorithm to retrieve the Watermark as text message ............................................... 25
   3.4.3 Pre-processing Before Embedding and After Retrieving .......................................... 26
   3.4.4 Enhanced Robustness Due to the Pre-processing ....................................................... 26
Chapter 4

ANALYSIS OF EXPERIMENT RESULTS

4.1 Conducted Experiment

4.2 Implementation

4.2.1 Tests for Computational Speed

4.2.2 Tests with various size of image

4.2.3 Tests with various size of message

4.2.4 Capacity of the message in an Image

4.2.5 How strong is the encryption based on different cases?

4.3 Discussion

4.4 Limitations

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

5.2 Future Work

Glossary

References
LIST OF FIGURES

Figure 1.1 Normal data flow................................................................. 2
Figure 1.2 Security Attacks................................................................. 2
Figure 2.1 Steganography vs. Cryptography........................................... 8
Figure 2.2 General model of Cryptography System............................... 8
Figure 2.3 Symmetric encryption.......................................................... 9
Figure 2.4 Asymmetric Encryption....................................................... 11
Figure 2.5 Pure steganography process............................................... 15
Figure 2.6 Secret key steganography................................................... 15
Figure 2.7 Public key steganography................................................... 15
Figure 2.8 General watermarking system.......................................... 17
Figure 3.1 System Model................................................................. 20
Figure 3.2 GUI for the system............................................................. 21
Figure 3.3 Canny Edge Detection....................................................... 23
Figure 3.4 LSB Substitution............................................................... 24
Figure 4.1 Experimentered Image....................................................... 28
Figure 4.2 Implementation Process 1................................................... 29
Figure 4.3 Implementation Process 2................................................... 29
Figure 4.4 Implementation Process 3................................................... 30
Figure 4.5 Implementation Process 4................................................... 30
Figure 4.6 Implementation Process 5................................................... 31
Figure 4.7 Implementation Process 6................................................... 31
Figure 4.8 Implementation Process 7................................................... 32
Figure 4.9 Implementation Process 8........................................................... 32
Figure 4.10 Implementation Process 9.......................................................... 33
Figure 4.11 Implementation Process 10....................................................... 33
Figure 4.12 Plotting values of different size of image (in bytes) and their PSNR values............................................................. 36
Figure 4.13 Plotting values of different size of image (in bytes) and their MSE values............................................................. 36
Figure 4.14 Plotting values of different size of message (in letters) and their PSNR and MSE values................................................. 38
Figure 4.15 Canny edge detection for small image and large image......... 39
Figure 4.16 MSE and PSNR values for low image and high image w.r.t (Key,Message).......................................................... 42
LIST OF TABLES

Table 2.1 Vigenere Square.......................................................... 12
Table 3.1 Vigenere Square with Numbers........................................... 22
Table 4.1 Experiments for showing the computational speed......................... 34
Table 4.2.1 Experiments with different size of the image with constant Message................................................................. 35
Table 4.2.2 Experiments with different size of the image with various methods ................................................................. 35
Table 4.3 Experiments with different size of the message with constant Image................................................................. 37
Table 4.4 Experiments for showing which case is best for strong encryption. 41
Chapter 1

INTRODUCTION

1.1 Motivation

Modern data hiding has a variety of uses but two of the primary fields are watermarking and steganography. Fundamentally, watermarking can be described as a method for embedding information into another signal. In case of digital images, the embedded information can be either visible or invisible (hidden) from the user. In my thesis, I will concentrate on hidden watermarks. Typical usage scenarios for watermarking are e.g. copyright protection and data authentication.

Another motivation for researching the topic was after reading an online article in the USA Today titled "Terror groups hide behind Web encryption" that claims terrorists and, in particular, Osama bin Laden and the al-Qaida network, may be using steganography to communicate with each other in planning terrorist attacks. It is thought that images with hidden messages are placed on bulletin boards or dead drops for other terrorists to pick up and retrieve hidden messages. Thus far, this supposition has yet to be proven.

1.2 Intellectual Property and the Digital Age

Rapid evolution of digital technology has improved the ease of access to digital information enabling reliable, faster and efficient storage, transfer and processing of digital data [1]. It also leads to the consequence of making the illegal production and redistribution of digital media easy and undetectable. Hence the risk of copyright violation of multimedia data has increased due to the enormous growth of computer networks that provides fast and error free transmission of any unauthorized duplicate and possibly manipulated copy of multimedia information. One way to protect multimedia data against illegal recording and distribution is to embed a secondary signal or pattern into the image, video or audio data that is not perceivable and is mixed so well with the original digital data that it is inseparable and remains unaffected against any kind of multimedia signal processing. This embedded secondary information is digital watermark which is, in general, a visible or invisible identification code that may contain some information about the intended recipient, the lawful owner or author of the original data, its copyright etc. in the form of textual data or image.

The information to be hidden is embedded by manipulating the contents of the digital data, allowing someone to identify the original owner, or in the case of illegal duplication of purchased material, the buyer involved. This digital watermark can be detected or extracted later to make an assertion about the data. Digital watermarks remain intact under transmission / transformation, allowing us to protect our ownership rights in digital form.
Absence of a watermark in a previously watermarked image would lead to the conclusion that the data content has been modified. In order to be effective for copyright protection, digital watermark must be robust, recoverable from a document, provide the original information embedded reliably, and be non-intrusive and also removable by authorized users.

### 1.3 Problem Statement

The data is transmitted from source to destination which is known as its normal flow as shown in the figure. But the hackers might hack the network in order to access or modify the original data. These types of attacks are formally known as security attacks.

![Figure 1.1 Normal data flow.](image)

A hacker can disrupt this normal flow by implementing the different types of techniques over the data and network in following ways. The security attacks are classified in four ways such as

![Figure 1.2 Security Attacks.](image)
**Interruption**- Interruption is an attack by which the hackers can interrupt the data before reaching the destination. This type of attack shows the effect on availability and usually destroys the system asset and makes the data unavailable or useless.

**Interception**- Interception is one of the well known attacks. When the network is shared that is through a local area network is connected to Wireless LAN or Ethernet it can receive a copy of packets intended for other device. On the internet, the determined hacker can gain access to email traffic and other data transfers. This type of attack shows the effect on confidentiality of data.

**Modification**- This refers to altering or replacing of valid data that is needed to send to destination. This type of attacks is done usually by unauthorized access through tampering the data. It shows effect on the integrity of the data.

**Fabrication**- In this type, the unauthorized user places data without the interface of source code. The hacker or unauthorized person inserts the unauthorized objects by adding records to the file, insertion of spam messages etc. This type of attack affects on the Authenticity of message.

There are many types of security attacks that will try to modify the original data. The main goal of any organization / individual transmitting the data is to implement security measures which include

**Prevention** - The security attacks can be prevented by using an encryption algorithm to restrict any unauthorized access to the encryption keys. Then the attacks on confidentiality of the transmitted data will be prevented.

**Detection** - Using the intrusion detection systems for detection of unauthorized individuals logged onto a system and making the resources available to legitimate users.

**Response** - Whenever the unauthorized attacks happen in the system, the security mechanisms can detect the process and the system can respond to make the data unavailable.

**Recovery** - Recovery is the final approach if an attacker modifies the data or makes the data unavailable. The data can then be recovered by using backup systems, so that the integrity of the data shall not be compromised.

### 1.4 Objective

Objective of this thesis is to develop or modify any existing image watermarking algorithm with steganography techniques as LSB, which is to hide a text of a secret message in the pixels of the image in such a manner that the human visual system is not able to distinguish.
between the original and the stego-image. My contribution for this proposed watermarking, as usual LSB technique but with the hint of embedding the message only on the image edges alone. Then while coming to cryptography algorithm, I have taken Vigenere square with extension of numbers also. So that we can use the key combination of characters and numbers. So by using these modifications in this existing algorithm and combination of cryptography and steganography method we develop a secure and strong watermarking method. While coming to see robust of the proposed encryption, if there is higher value of PSNR with small value of MSE then it seems to be strong.

1.5 Proposed solution to prevent the Security Attacks

There are different types of approaches for preventing the security attacks. The most useful approaches are

1. Cryptography
2. Steganography
3. Digital watermarking

“Cryptography” is the art of science used to achieve security by encoding the data to transform them into non readable formats so that unauthorized users cannot gain access to it. The encoded text is known as ”Cipher text” and this technique is known as encryption and this process is reversed with authorised access using the decryption technique, in which the encoded data is decoded into readable format [2]. While Cryptography is a method to conceal information by encrypting it to ”cipher texts” and transmitting it to the intended receiver using an unknown key. Steganography provides further security by hiding the cipher text into a seemingly invisible image or other formats.

According to Johnson et al, [3] “Steganography” is the art of hiding and transmitting data through apparently innocuous carriers to conceal the existence of data. The level of visibility is decreased using many hiding techniques in Image Modelling like LSB Manipulation, Masking and filtering. These techniques are performed by different steganographic algorithms like F5, LSB, Hide and Seek etc. and the act of detecting the information hidden through these algorithms is called Steganalysis.

Steganography and Cryptography are closely related constructs. The hidden or embedded image, audio or a video files act as carriers to send the private messages to the destination without any security breach. Steganography techniques can be implemented on various file formats such as audio (mp3, wmv etc.), video (mpeg, dat, etc.) and images ( jpeg, bmp etc.). However, the images are the most preferred file format for this technique. At present, there are a lot of algorithms that help in executing the steganography software [4].
“Digital watermarking” is described as one of the possibilities to close the gap between copyright issues and digital distribution of data. It is mainly based on Steganographic techniques and enables useful safety mechanisms [5]. It acts as a very good medium for copyright issues as it embeds a symbol or a logo in the form of a watermark, which cannot be altered manually. One critical factor to be kept in mind when using steganography is to prevent any further alterations to the originality of the image after embedding the data. Whenever the image with the secret data is transmitted over the internet unauthorized parties may want to hack the data hidden over the image. So, if the originality of the image has been changed then it will be easier to hack the information by unauthorized persons. In order to improve the security, the Digital watermarks are predominantly inserted as transformed digital signal into the source data using key based embedding algorithm and pseudo noise pattern.

This technique has also found big use in the notorious hands of terrorists and the September 2001 Twin tower attacks of the USA are predominantly associated with the communications using steganography. The Steganalysis aims at discovering and decrypting the suspected data transferred with the use of the available algorithms.

1.6 Content of the Project

Chapter-1: Introduction: In this section, the main points discussed are about the Overview, the Background and Objective of the project, the motivation of the project, problems of various security attacks and how to prevent them with the 3 techniques and the approach to research employed are discussed.

Chapter-2: Literature Review: Definitions and overview about the different information security methods to gather knowledge on the existing theories of steganography and review it for proposing an improvised system for providing the required security and discuss about different functionalities of algorithms used for the proposed system. Evaluation of PSNR(Peak Signal to Noise Ratio) and MSE (Mean Square Error) values.

Chapter-3: Proposed Watermarking Scheme: It says about the extension of cryptography algorithm and LSB (Least Significant Technique) substitution in the image for embedding the message using Steganography.

Chapter-4: Analysis of experiment results: The experiment is done with various size of image and also with various size of message. Implementation of different modules like encryption, decryption and data hiding techniques. It also discusses about the PSNR (Peak Signal to Noise Ratio) and MSE (Mean Square Error) values.

Chapter-5: Conclusion and Future work: Here, the project is concluded with the results of the proposed method that has been analysed and recommendations are made according to the results obtained from the analysis.
Chapter 2

THEORETICAL BACKGROUND

In the last few years, numerous schemes have been developed for image watermarking system to hide a digital text of a secret message using different types of Cryptography and Steganography algorithms. LSB based algorithms is one of the simplest algorithms with very high data rate of additional information. The previous work of the image steganography is discussed as follows.

2.1 Literature Review

Yu-Chee Tseng (et al) [6] proposed statistical based scheme uses a binary matrix and an integer weight matrix as secret keys. The operator XOR is adopted so that the keys cannot be compromised easily. The proposed statistical scheme uses the weight matrix to represent the embedded data. Nasir Memon (et al) [7] proposed block based partitioning both host and binary watermark image into blocks, setting LSB’s of each image block to zero, applying hash function (MD5) [8] to image block. Li Zhi (et al) [9] proposed steganography by using smooth characteristics between adjoining pixels of the image. The relation between the length of embedded message and the gradient energy is theoretically analyzed. The performance is named as gradient energy-flipping rate detection [10] (GEFR). Pal, A.J (et al) [11] proposed audio steganography by using two level of security, one is encryption algorithm (RSA) to encrypt message and the other one is GA (Genetic Algorithm) based LSB algorithm to encode the encrypted message into audio data. Mohammed A.F. (et al) [12], proposed a new framework of an image steganography by using 7 Most Significant Bits (MSBs). B.Macq (et al) [13] proposed feature based to find feature points and apply Delaunay tessellation to obtain the triangular sequence. Detection is performed by finding Delaunay tessellation [14] of the test image and wiener filtering to obtain watermark and then performing correlation. Dr V.V.Krishna (et al) [15], proposed a method related to LSB watermarking whereas any image is represented by a two dimensional array and divides the image into non overlapped window of a predefined size. While embedding the watermark the order of selection of the hit pixels in the window is based on the least co-ordinate position.
2.2 Data Hiding Techniques

There are several techniques for information hiding into digital media. They are used for several purposes as well as copyright protection. Two basic methods of information hiding are cryptography and steganography. The concept of digital watermarking is derived from steganography. The term steganography means “cover writing” and cryptography means “secret writing”. Cryptography is a widely used method for protecting the digital content of the media. The message is encrypted before transmission and decrypted at the receiver end with the help of a key. No one can access the content without having the true key. The message is called the plain text and the encrypted message is called the cipher text [16]. The information is protected before the time of transmission. But, after decryption, the information becomes unprotected and it can be copied and distributed. The schematic representation of the cryptography is given in Figure 2.1 (b).

In steganography, the message is embedded into the digital media rather than encrypting it in such a way that nobody except the sender and the intended recipient can even realize that there is a hidden message. The digital media content, called the cover, can be determined by anybody; but the message hidden in the cover can be detected by only the person having the actual key. Thus steganography actually relates to covering point-to-point communication between two parties. That’s why steganography methods are usually not robust against modification of the data, or have only limited robustness. The schematic representation of the steganography is given in Figure 2.1 (a).

![Steganography vs. Cryptography](image)

**Figure 2.1 Steganography vs. Cryptography [16].**

**Steganography**- Steganography security hides the “knowledge” that is information in the cover medium.

**Cryptography**- Cryptography reveals this “knowledge” but encodes the data as cipher text and disputes decoding it without permission. (i.e.) It concentrates on the challenge on the decoding process while steganography adds the search for detecting if there is hidden information or not.
2.3 Cryptography

The word cryptography is derived from two Greek words which mean “secret writing”. Cryptography is the process of scrambling the original text by rearranging and substituting the original text, arranging it in a seemingly unreadable format for others. Cryptography is an effective way to protect the information that is transmitting through the network communication paths [17].

Cryptology is the science that deals about cryptography and cryptanalysis. Cryptography is the approach of sending the messages secretly and securely to the destination. Cryptanalysis is the method of obtaining the embedded messages into original texts [18]. In general, cryptography is transferring data from source to destination by altering it through a secret code. The cryptosystems uses a plaintext as an input and generate a cipher text using encryption algorithm taking secret key as input.

![Figure 2.2 General model of Cryptography System [19].](image)

The important elements in cryptography systems are plain text, secret key, cipher text, encryption and decryption algorithm. The plain text is an original piece of information that is needed to send information to the destination. The secret key is given by the user which will act as an input to the encryption algorithm. Based on this key, various substitutions and transformations on the plain text will differ. This is the output generated by the encryption algorithm. The cipher text is the jumbled text. The cipher text differs with each and every secret key that has given to the encryption algorithm. The encryption algorithm is the main key to any cryptographic system. This encryption algorithm subjects the plain text to various substitutions and transformations. Decryption algorithm is opposite to the encryption algorithm. It will acquire cipher text and secret key as an input and produce plain text as an output.

2.4 Cryptography Algorithms

There are many cryptographic algorithms available which differ on their type of encryption. Based on the type of encryption standards the algorithms are grouped into two types
2.4.1 Symmetric Encryption

Symmetric encryption is a single key encryption and also known as conventional encryption. It is also referred to “private key cryptography”. The symmetric encryption algorithm generally uses the same key for “encryption” and “decryption”. The security level for this type of encryption will depend on the length of the key.

There are two types of methods that will attack on symmetric encryption systems. The first one is Cryptanalysis. If the attacker gets to know some information about the plain text and cipher text, he analyses the characteristics of the algorithms used for encryption and tries to generate keys. The second type of attack is known as “brute force attack”. In this type of attack, the defender attempts to know the cipher text and try every possible key for translation. To avoid this problem, the user should use the key that no longer can be estimated like 128 or 168 bit keys [21].

Block cipher is an asymmetric algorithm in which the cipher processes the text in fixed size blocks and generates same size cipher text blocks. In this algorithm, the plaintext is divided into independent blocks of 8-16 bytes and encrypts each block independently.

The different symmetric encryption algorithms are Data encryption standard, Triple DES and Advanced encryption standard. Data Encryption Standard (DES) is also known as Data Encryption Algorithm (DEA). DEA takes 64 bits of plain text and 56 bits of key to produce 64 bits cipher text block. The DES algorithm always functions on blocks of equal size and uses the permutations and substitutions in algorithm. The data encryption algorithm uses 56 bit key so it is not possible for the defender for analyzing the key. So, the problem of Cryptanalysis is avoided using this algorithm. But the drawback of the algorithm is Brute-force attack. This can be avoided using the Triple DES algorithm.
Triple DES is an extension to the DES algorithm. Triple DES uses the same approach for encryption as DES. 3DES takes three 64 bit keys which has a total length of 192 bits. We can give more than one key that is two or three keys for encryption as well as for decryption such that the security will be stronger. It is approximately 256 times stronger than the normal DES algorithm, so that this algorithm can avoid the brute force attack. The main drawback of using 3DES algorithm is that the number of calculations is high reducing the speed to a greater extent. And the second drawback is that both DES and 3DES use same 64 block size to avoid security issues. “Advanced Encryption Standard” algorithms are used to avoid these limitations.

Advanced Encryption Standards (AES) takes a block of size 128 bits as input and produces the output block of same size. AES supports different key sizes like 128, 192 and 256 bit keys. Each encryption key size will change the number of bits and also the complexity of cipher text. The major limitation of AES is error propagation. The encryption operation and key generation both engage in number of non linear operations, so, for lengthy operations it is not suitable. A cryptanalyst may able to use the continuities in plain text to simplify the decryption [18].

2.4.2 Asymmetric Encryption

Asymmetric encryption is also known as Public key encryption. The AES works same as Symmetric encryption, the main difference between AES and Symmetric encryption is in using keys. In asymmetric encryption, the encryption and decryption will be done by two different keys. It will use plain text, encryption algorithm and decryption algorithm same as Symmetric encryption as discussed in above section.

In Asymmetric encryption, only the data that is encrypted using public key can be decrypted using the same algorithm. And the message which is encrypted using private key can be decrypted using only the matching public key. The main problem with Asymmetric algorithm is “cipher keys”. Whenever two different people want to exchange the data simultaneously using asymmetric encryption they need to have four different keys. It will be more confusing to resolve as the corresponding key is required for the particular file to open.
The most important public key encryption algorithm is RSA algorithm. It was first developed in 1977. RSA functions depend upon the large prime numbers of public and private keys. The security is also based on the difficulty of prime numbers. The RSA algorithms are used in public key encryptions as well as in digital signatures. It allows the sender to encrypt the message using public key and decrypt the message using private key by receiver. So, the security will be high using RSA in public key encryption [23].

2.4.3 Vigenere Square

Generally we will see how the Vigenere Square or Vigenere Cipher Square works in the encoding and decoding process. Vigenere Square is to applicable when both encoder and decoder has the same key.
Table 2.1 Vigenere Square [24].

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
| Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |

2.4.3.1 Encoding

To see how it works, it's best to give it a try. So, let's code the sentence “MEET ME AFTER SCHOOL”. First, write out the sentence without the spaces in between and write the keyword below it, repeating the characters until it is as long as the sentence you are encoding.
Then for each combination, find the character that is on the intersection of the column (top character) and the row (bottom character). To get the coded character for the top letter (the first letter of the sentence "M") you go down the rows until you reach the row that has the bottom character (the first letter of the word ESPIONAGE, "E"). The character that's on the intersection is "Q".

![Table]

Okay, one more time. Top character is "E", bottom character is "S". Going down the column "E", until you reach row "S", you find the coded letter "W" and so on.

![Table]

### 2.4.3.2 Decoding

Now what do you do when you receive that message? Well, it is more of the same, just the other way around. Write the coded text, and below it the keyword, repeating as long as the coded text is.

![Table]

Now for each combination, do the following. Find the row for the bottom character. In our example, "E" (first letter of "ESPIONAGE"). Then looks through that row until you find the character in the top row, in this case "q". Then go up to see the letter that is at the top of that column. The column that has the letter "q" is the "M" column. Next character to look for is the letter "w" in the row "S". The letter "w" on that row can be found in the column "E" and so on.

![Table]
2.5 Steganography

Steganography in Greek means “covered writing”. Steganography is the process of hiding the one information into other sources of information like text, image or audio file, so that it is not visible to the natural view. There are varieties of steganographic techniques available to hide the data depending upon the carriers we use.

Digital steganography refers to the practice of altering an innocuous looking file to contain a secret message so that an observing adversary would be unaware that the secret message is being delivered, but only the innocent-looking file. This has diverse military and espionage purposes in the current global environment.

Steganography includes different methods for hiding the existence of additional information in an innocuous signal. The distinction between steganography and watermarking is not always clear. Basically, in case of watermarking the additional information is used to protect the original image (e.g. in case of copyright management), whereas in the steganography the image is used to protect the additional information (e.g. secret message). By definition, the visible watermarks are not included in steganography. We will go through information hiding, where a ‘secret’ message is hidden in the least significant bits of a original RGB image.

Steganography and cryptography both are used for the purpose of sending the data securely. The same approach is followed in Steganography as in cryptography like encryption, decryption and secret key. In steganography the message is kept secret without any changes but in cryptography the original content of the message is differed in different stages like encryption and decryption.

Steganography supports different types of digital formats that are used for hiding the data. These files are known as carriers. Depending upon the redundancy of the object the suitable formats are used. “Redundancy” is the process of providing better accuracy for the object that is used for display by the bits of object. The main file formats that are used for steganography are Text, images, audio, video, protocol [25].

The different types of steganographic techniques that is available are
1. Pure steganography
2. Public key steganography
3. Secret key steganography

**Pure steganography**: Pure steganography is the process of embedding the data into the object without using any private keys. This type of steganography entirely depends upon the secrecy. This type of steganography uses a cover image in which data is to be embedded, personal information to be transmitted, and encryption decryption algorithms to embed the message into image.
This type of steganography can’t provide the better security because it is easy for extracting the message if the unauthorized person knows the embedding method. It has one advantage that it reduces the difficulty in key sharing [26].

**Secret key steganography:** Secret key steganography is another process of steganography which uses the same procedure other than using secure keys. It uses the individual key for embedding the data into the object which is similar to symmetric key. For decryption it uses the same key which is used for encryption.

This type of steganography provides better security compared to pure steganography. The main problem of using this type of steganographic system is sharing the secret key. If the attacker knows the key it will be easier to decrypt and access original information.

**Public key steganography:** Public key steganography uses two types of keys: one for encryption and another for decryption. The key used for encryption is a private key and for decryption, it is a „public key“ and is stored in a public database [26].
For encryption and decryption of text messages using the secret keys steganographic system uses algorithms known as steganographic algorithms. The mostly used algorithms for embedding data into images are LSB technique.

### 2.6 LSB Technique

LSB (Least Significant Bit) substitution is the process of adjusting the least significant bit pixels of the carrier image. It is a simple approach for embedding message into the image. The Least Significant Bit insertion varies according to number of bits in an image. For an 8 bit image, the least significant bit i.e., the 8th bit of each byte of the image is changed to the bit of secret message. For 24 bit image, the colours of each component like RGB (red, green and blue) are changed. LSB is effective in using BMP images since the compression in BMP is lossless. But for hiding the secret message inside an image of BMP file using LSB algorithm it requires a large image which is used as a cover.

LSB substitution is also possible for GIF formats, but the problem with the GIF image is whenever the least significant bit is changed the whole colour palette will be changed. The problem can be avoided by only using the gray scale GIF images since the gray scale image contains 256 shades and the changes will be done gradually so that it will be very hard to detect. For JPEG, the direct substitution of steganographic techniques is not possible since it will use loss compression. So it uses LSB substitution for embedding the data into images. There are many approaches available for hiding the data within an image: one of the simple least significant bit submission approaches is "Optimum Pixel Adjustment Procedure". Amirtharajan (et al.) [27] for OPA explains the procedure of hiding the sample text in an image. This method of substitution is simple and easy to retrieve the data and the image quality better so that it provides good security.

### 2.7 Digital Watermarking

Digital watermarking is of high priority with increasing use of digital media and increasing conflict concerning copyright infringement and piracy. A secure watermarking algorithm would allow creators of digital property to prove their ownership of files. It would also allow for tracking of files in controlled distribution, resulting in better control of piracy.

During the past years, digital watermarking has attracted the attention of numerous researchers. As a result, hundreds of studies have been published concerning the different methods for watermarking. The information embedded as a watermark can be almost anything. It can be a bit string representing copyright message, serial number, plain text, etc. However, sometimes it can be more useful to embed a visual watermark (e.g. corporate logo) instead of a bit string as a watermark.
Watermarking aim is to protect the cover medium from any modification with no real emphasis on secrecy. It can be observed as steganography that is concentrating on high robustness and very low or almost no security. Image Watermarking as mentioned earlier is the process of embedding a secondary signal into an image such that the signal can be detected or extracted later to make an assertion about the image. In general, any watermarking scheme consists of the following three parts namely

- The watermark signal,
- Watermark embedded that embeds the watermark into the media.
- Watermark detector that verifies the presence of watermark.

![Watermarking System Diagram](image)

**Figure 2.8 A typical watermarking system [28].**

Figure 2.8 is a conventional watermarking system [28] consists of watermark embedded and watermark detector. The inputs to the watermark embedded are the watermark, the cover media data and the embedding security key. The watermark can be a number sequence, a binary bit sequence or may be an image. The key is used to enhance the security of the whole system. The output of the watermark embedded is the watermarked data. The inputs to the watermark detector are the watermarked data, the security key and, depending on the method, the original data and/or the original watermark.

There are two type of watermarking techniques one is robust watermarking and another is fragile watermarking. Robust watermarking is mainly used for the purpose of copyright protection because they are strong for all kinds of manipulations in images. The second method fragile watermarking is used for providing better authentication and for verification of integrity in order to avoid the modifications [29].
The applications of watermarking are:

**Copyright protection**: Watermarks are used for copyright protection by embedding the watermark secretly which can be read only through the secret key held by the owner.

**Monitoring**: Watermarks are used for tracing the illegal copying.

**Fingerprinting**: In the „point to point distribution” environments, the information on the authenticated customers could be embedded into secret watermarks well before the secure delivery of the data [30].

**Content manipulation indication**: The indication of content manipulation from the authorized state can be detected only by means of a public or fragile watermark.

**Information carrier**: A public watermark is embedded into the data stream that shall act as a link to the external databases to store information about the copyright and license conditions.

### 2.8 Computing the PSNR and MSE

PSNR is nothing but Peak Signal to Noise Ratio for showing the ratio between maximum possible power of signal and the power of corrupting noise that affects the image pixel. The PSNR is a common measure of the quality of a reconstructed digital sequence. After generating there constructed digital image sequence, the quality is analyzed using the PSNR measure. A logarithmic scale is used to compute the PSNR because the signals have a wide dynamic range. PSNR mostly ranges 70 – 90, when the PSNR is high then the quality of the reconstructed watermark image is “Good” and vice versa.

The PSNR formula is as follows [31]:

\[
\text{PSNR} = 10 \log_{10}(\frac{\text{max}^2 - 1}{\text{MSE}}).
\]

Where “max” denotes maximum possible value of an image.

The mean square error (MSE) of two images of \(H\times W\) pixels are defined as [31]:

\[
\text{MSE} = \frac{1}{H\times W} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (\text{Orig}(i,j) - \text{Recon}(i,j))^2.
\]

\(\text{Orig}(i,j)\) is the original pixel value and \(\text{Recon}(i,j)\) is the reconstructed pixel value. The higher the pixel value the better the quality of the reconstructed image. MSE values should not exceed to 1, if so the reconstructed watermark image is “Bad”. So the values should be less than 1, however if MSE value is very less than the reconstructed watermark image is “Good” respectively.
Chapter 3

PROPOSED WATERMARKING SCHEME

This chapter discusses the developed watermarking schemes. First, short background of the proposed scheme has been presented and then the detailed scheme has been described.

3.1 Developed Watermarking Scheme

My contribution for this proposed watermarking, as usual LSB technique but with the hint of embedding the message only on the image edges alone. Then while coming to cryptography algorithm, I have taken Vigenere square with extension of numbers also. So that we can use the key combination of characters and numbers. So by using these modifications in this existing algorithm and combination of cryptography and steganography method we develop a secure and strong watermarking method. While coming to see robust of the proposed encryption, if there is higher value of PSNR with small value of MSE then it seems to be strong.

While reviewing number of watermarking schemes based on LSB, it has been noticed that some algorithms are used for different types of spatial domain for embedding watermark. In this developed algorithm watermark is embedded in spatial domain with some pre-processing before and after the LSB technique. In all spatial domain watermarking schemes, there is a conflict between robustness and transparency. If the watermark is embedded in perceptually most significant components, the scheme would be robust to attacks but it would be difficult to hide the watermark. On the other hand, if the watermark is embedded in perceptually insignificant components, it would be easier to hide the watermark. This watermarking scheme uses LSB and Vigenere Square as Cryptography algorithm as pre-processing for embedding watermark. The watermark is a colour image. This is a destination based algorithm i.e. the cover image as Cipher Text. Before getting into LSB with Vigenere Square, we are dealing the watermark as text in our thesis by using Least Significant Bit (i.e) LSB based Steganography to hide the text behind the image and retrieve the text from watermarked image.
Initially, we select the plain text which is to be encrypted, plain text is an original piece of information that is needed to send information to the destination. The cryptography algorithm is applied with the Crypto key as C_key then the plain text is changed to Cipher text. The cipher text is the jumbled text. The cipher text differs with each and every crypto key (C_key) that has given to the encryption algorithm as LSB embedding technique which produces the stego key (S_key). Then the text is embedded into the image which is said to be watermarked image whereas here the text which is hidden in the image is said to be watermark. Decryption algorithm made by LSB extracting technique with Stego key (S_key) which acquire cipher text and then crypto key (C_key) as an input and produce stego text as an output.

The GUI is created for both encryption and decryption module in the same screen. The encryption module of steganography is the primary stage. In this stage, the sender sends the data as well as the image file with crypto key and produced stego key which act as a carrier image to transfer the data to destination. In this project, I use bit map (.bmp) images as carriers because bmp images are highly resistant for Steganalysis compared to jpeg images. In the encryption module, the text message will be embedded into the image file. The embedding will be done based on the principle of Least Significant Bit (LSB) algorithm. The LSB algorithm uses the least significant bits of each pixel and replace with the significant bits of the text document, such that the message will be encrypted into the image. This process makes the picture not to lose its resolution. The data embedding into image i.e. encryption is implemented using MATLAB. The encrypted data is send to the receiver or authorized person with the help of transmission media for example through web or E-Mail. The image in which the data is embedded acts as a carrier file such that the data can be transmitted easily with high security.
Using the Least Significant Bit (LSB) algorithm the message bits can be embedded properly in the place of least significant bits of image, such that the image doesn’t lose its resolution. So, the security will be high. The encrypted image is protected with both stego key and crypto key such that we can avoid the damages caused due to hackers or unauthorized persons.

In the decryption module, the receiver receives the carrier image from sender through the transmission medium. The receiver then sends the carrier image to the decryption phase. In the decryption phase, the same Least Significant Algorithm (LSB) is implemented for decrypting the least significant bits from the image and merge in an order to frame the original message bits. After successful arrangement, the file is decrypted from the carrier file and accessed as an original text document. The data extraction from the image that is decryption is implemented by giving correct S_key and C_key respectively. If the stego key (S_key) is wrong then my MATLAB file gives error dialog that you have entered the wrong key and we have another chance to attempt for retrieving the message, it goes up to 3 chance if not giving correct key in 3 chance then the whole file is closed. In case of crypto key (C_key) even though the key is wrong it gives some message but not the correct message that makes the hacker to get confuse and speciality in our system.

As for showing the results, the GUI is created as follows:

![GUI for the system](image.png)

**Figure 3.2 GUI for the system.**
3.2 Cryptographic Algorithm

If the hacker comes to know to know whether the image is going to handle with Steganography then it retrieves the message easily. So we handle the message with Cryptography algorithm of Extension of Vigenere Square before embedding into the image with the Steganalysis by using LSB technique.

3.2.1 Extension of Vigenere Square

Currently many crypto forums is discussing about the extension of Vigenere Square for more secured encryption [32]. So i have implemented the extended Vigenere Square with numbers and lower and upper case characters of English alphabets.

Table 3.1 Vigenere Square with Numbers.

<table>
<thead>
<tr>
<th>a</th>
<th>B</th>
<th>......</th>
<th>Z</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>......</th>
<th>9</th>
<th>0</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>a</td>
<td>B</td>
<td>......</td>
<td>Z</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>......</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>C</td>
<td>......</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>......</td>
<td>0</td>
<td>Space</td>
</tr>
<tr>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
</tr>
<tr>
<td>Z</td>
<td>z</td>
<td>1</td>
<td>......</td>
<td>N</td>
<td>o</td>
<td>p</td>
<td>q</td>
<td>......</td>
<td>w</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>......</td>
<td>O</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>......</td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>......</td>
<td>P</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>......</td>
<td>y</td>
<td>z</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>......</td>
<td>Q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>......</td>
<td>z</td>
<td>1</td>
</tr>
<tr>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
<td>.......</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>0</td>
<td>......</td>
<td>W</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td>......</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Space</td>
<td>......</td>
<td>X</td>
<td>y</td>
<td>z</td>
<td>1</td>
<td>......</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Space</td>
<td>Space</td>
<td>A</td>
<td>......</td>
<td>Y</td>
<td>z</td>
<td>1</td>
<td>2</td>
<td>......</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>
3.3 Steganography Method

In Steganalysis, I have implemented the LSB technique with clear substitution. By using the LSB technique for edges of the image such as it is easy to retrieve from the image while decoding it.

3.3.1 Canny Edge Detection

Edge detection is a fundamental tool in image processing, mainly in the areas of feature detection and extraction. Edges are the most common feature in the image and also local variation function in the image. It aims at identifying points in a digital image at which the image brightness changes sharply and has discontinuities. The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection". In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be NO responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Canny saw the edge detection problem as a signal processing optimization problem, so he developed an objective function to be optimized. The solution to this problem was a rather complex exponential function, but Canny found several ways to approximate and optimize the edge-searching problem. The steps in the canny edge detector are as follows:

![Canny Edge Detection](image-url)

Figure 3.3 Canny Edge Detection[33].
1. Smooth the image with a two dimensional Gaussian. In most cases the computation of a two dimensional Gaussian is costly, so it is approximated by two one dimensional Gaussians, one in the x direction and the other in the y direction.
2. Take the gradient of the image. This shows changes in intensity, which indicates the presence of edges. This actually gives two results, the gradient in the x direction and the gradient in the y direction.
3. Non-maximal suppression. Edges will occur at points where the gradient is at a maximum. Therefore, all points not at a maximum should be suppressed. In order to do this, the magnitude and direction of the gradient is computed at each pixel. Then for each pixel check if the magnitude of the gradient is greater at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient. If the pixel is not greater than both, suppress it.
4. Edge Thresholding. The method of thresholding used by the Canny Edge Detector is referred to as "hysteresis". It makes use of both a high threshold and a low threshold. If a pixel has a value above the high threshold, it is set as an edge pixel. If a pixel has a value above the low threshold and is the neighbour of an edge pixel, it is set as an edge pixel as well. If a pixel has a value above the low threshold but is not the neighbour of an edge pixel, it is not set as an edge pixel. If a pixel has a value below the low threshold, it is never set as an edge pixel.

3.3.2 LSB Substitution Techniques

For substituting the message by LSB technique we are finding the edges in the image. While retrieving the text it cross check the edges of the image and their corresponding pixel value of it. Just form the array for the edges pixel value of the LSB which is binary. Then arrange them for 8-bit which is binary and converting them as character to decode the message. For example, if you need to embed the text as “A” which is binary equivalent to **01000001** whereas a1=0, a2=1, a3=0, a4=0, a5=0, a6=0, a7=0 and finally a8=1. Likewise the text is embedded in the image of the corresponding pixels LSB value.

The below diagram shows how the LSB substitution works in our algorithm by using Canny Edge Detector.

**Figure 3.4 LSB Substitutions.**
3.4 LSB based Steganography

3.4.1 Algorithm to embed the Watermark as text message

1) Get the original (colour) image as bitmap format.
2) Enter the plain text as string.
3) Find the object tag as message in the GUI and get its string value where we save the plain text.
4) Change the string value to its equivalent binary value.
5) Change the rgb (colour) image to gray image for finding the image edges easily.
6) Taking the edges of 8 consecutive pixel binary values for saving the binary equivalence of the text message which is to be hidden.
7) In this image Steganalysis method, binary equivalent of the message (to be hidden) is distributed among the LSBs of each pixel.

3.4.2 Algorithm to retrieve the Watermark as text message

1) Read the watermarked (stego) image, with the watermark as text.
2) Finds the object tag as Cipher Text in the GUI and get its string value.
3) Convert the string to number for getting their length which is the stego key.
4) Change the rgb(colour) image to gray image for finding the image edges easily.
5) It checks and decodes all the edge pixel of the Least Significant bit alone.
   5.1) It gets the pixel value of the each edge pixel (in decimal) value.
   5.2) Change the decimal value to binary value of the Least Significant Bit alone.
   5.3) finally, we change the binary values to the characters.
6) Retrieved the message successfully without the help of Original image.
3.4.3 Pre-processing Before Embedding and After Retrieving

This step enhances the robustness against the secured authentication of the watermarked image. This pre-processing requires following tasks to be followed:

1. Get the size of the plain text which is to be embedded and apply Vigenere Square in the plain text.

2. Thus plain text is not entering into the watermarked image, whereas Cipher text is embed with crypto key.

3. While retrieving the text from watermark image, we won’t get the stego text.

4. The hacker doesn’t know the retrieved text is Cipher text.

5. We apply cryptography algorithm as Extension of Vigenere Square with numbers to get the stego text.

3.4.4 Enhanced Robustness Due to the Pre-processing

Applying Extension of Vigenere Square with numbers to plain text adds the extra key as crypto key with the stego key. So if both keys are correct only we can retrieve the original plain text. This makes the algorithm and the system strong while using the Extension of Vigenere Square with numbers for the LSB watermarking and results in the successful extraction of the watermark image.
Chapter 4

ANALYSIS OF EXPERIMENT RESULTS

This chapter is for describing the conducted experiment and evaluating the values of their PSNR (Peak Signal to Noise Ratio) and MSE (Mean Square Error) of the experimented images. For seeing the quality of reconstruction of the original image which we receive after the encoding process by the Cryptographic and Steganography algorithms.

4.1 Conducted Experiment

The proposed algorithm of the image steganography system is tested by taking different messages of different length and hiding them in the image and also taking different size of the image with constant message length by various edge detection operator as canny (with & without noise) and log (Laplacian of Gaussian) operator. According to see computation power, whereas which is the fast method to embed and retrieve the message from the image with respective time (in seconds). The experiment is done for both symmetric edge operators such as canny and log with both noise and without noise. Finally, I have took the best (high) and worst (low) case for my major 3 aspects such as encryption key, size of image and size of message. The experimented images are as follows.

4.1 (a) Image for Computing Power. 4.1 (b) Large image with less edges.
4.1 (c) Small image with more edges. 4.1 (d) Image for Various Message.

4.1 (e) Images for Constant Message.

Figure 4.1 Experimented Image.
4.2 Implementation

In encoding process, first type the message which is to be encrypted, and also type the crypto key then select the image so that encryption is done and it produces stego key. While decoding process just we need both the produced stego key and crypto key which gives you decrypted message respectively.

For example, we will type the message as “Hello Sweden” and the crypto key as ”son” as shown below.

![Figure 4.2 Implementation Processes 1.](image1)

Now click the ”SELECT IMAGE” button to select the image file to be opened from the path.

![Figure 4.3 Implementation Processes 2.](image2)

After selecting the image the "ENCRYPTION" button is ready in GUI, with the Original image in the other MATLAB figure window and thus it also produce the stego key as ”12” for the image in the GUI.

Dalarna University
Röda vägen 3S-781 88
Borlänge Sweden
Tel: +46(0)23 7780000
Fax: +46(0)23 778080
http://www.du.se
After that just click the "ENCRYPTION" button to do the encoding process. The below figure shows the Original image and Watermarked image whereas the text is encrypted as watermark in the image.
Figure 4.6 Implementation Processes 5.

Now the decoding process starts, just giving the produces stego key as "12" and crypto key as "son" which we give while encrypting the message.

Figure 4.7 Implementation Processes 6.
Just click the "DECRYPTION" button, so that it checks whether the entered 2 keys are same, if so it just does Canny edge detection to the watermark image and gives the decrypted message.

![Image](image.png)

**Figure 4.8 Implementation Processes 7.**

![Image](image2.png)

**Figure 4.9 Implementation Processes 8.**
If we enter the wrong key among the stego and crypto key, then the system gets failure. In case of stego key is wrong, even though the crypto key is correct. Like in our example, the produced stego key as “11” (incorrectly) and the correct crypto key as “son”. Then the system shows the new dialog box as error which shows the message as “You have entered the wrong key, please try again”.

![Image of stego and crypto key process]

**Figure 4.10 Implementation Processes 9.**

Likewise in case of crypto key is wrong, even though the stego key is correct. Like in our example, the produced stego key as “12” (correctly) and the incorrect crypto key as “man”. Then the system does the decryption process, but the decrypted message will not be correct.

![Image of stego and crypto key process]

**Figure 4.11 Implementation Processes 10.**

So that we need to get the correct produced stego key for decryption and also the correct crypto key, thus shows the output correctly while the decoding process takes place.

---

Dalarna University  Tel: +46(0)23 7780000  
Röda vägen 3S-781 88  Fax: +46(0)23 778080  
Borlänge Sweden  
http://www.du.se
4.2.1 Tests for Computational Speed

The performance of the proposed dual process system works well in a same domain say, symmetric encryption with high computational speed and less time. Moreover it is secured and robust which is the main goal of the thesis. But if they are implemented in different domain as two individual processes symmetric encryption, then it takes too much time and moreover the data can be stolen by the unauthorized users. It is not secured and robust, thus the system model which is proposed are most robust and secured. The below table 4.1 shows that computational time is good (by using figure 4.1(a)) in our proposed symmetric encryption method compared with individual two process symmetric encryption respectively.

Table 4.1 Experiments for showing the computational speed.

<table>
<thead>
<tr>
<th>Method</th>
<th>Status of the image</th>
<th>Proposed Dual Process Symmetric Encryption</th>
<th>Two Individual Process Symmetric Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Embedded Time (in sec)</td>
<td>Recovered Time (in sec)</td>
<td>Embedded Time (in sec)</td>
</tr>
<tr>
<td>Canny</td>
<td>With Noise</td>
<td>2.09</td>
<td>2.48</td>
</tr>
<tr>
<td></td>
<td>Without Noise</td>
<td>2.16</td>
<td>3.17</td>
</tr>
<tr>
<td>Log</td>
<td>With Noise</td>
<td>1.89</td>
<td>2.66</td>
</tr>
<tr>
<td></td>
<td>Without Noise</td>
<td>1.76</td>
<td>2.56</td>
</tr>
</tbody>
</table>

By seeing the above table 4.1, we came to know our proposed method is good in computational speed compared to individual symmetric encryption. But according to edge detection method, canny is slightly lower than log method, whereas canny method is time consuming and log method is not so. Thus log method gives good result when we need the encryption is to be delivered within short span of time. But canny method is good, when we need large data (message) to be encrypted in the image which we are going to see next.
4.2.2 Tests with various size of image

The results that are got from the experiment of different size of the image with constant message length are recorded and summarized in the Table 4.2.

**Table 4.2.1 Experiments with different size of the images with constant message.**

<table>
<thead>
<tr>
<th>Exp. #</th>
<th>Size of Image (in bytes)</th>
<th>Proposed Two Individual Process (AUTHOR)</th>
<th>MSE</th>
<th>PSNR</th>
<th>MSE</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>196,662</td>
<td>0.0028</td>
<td>0.0030</td>
<td>78.4213</td>
<td>0.0030</td>
<td>78.1317</td>
</tr>
<tr>
<td>2</td>
<td>270,054</td>
<td>0.0024</td>
<td>0.0027</td>
<td>80.4649</td>
<td>0.0027</td>
<td>80.0399</td>
</tr>
<tr>
<td>3</td>
<td>480,054</td>
<td>0.0019</td>
<td>0.0021</td>
<td>83.9167</td>
<td>0.0021</td>
<td>82.7437</td>
</tr>
<tr>
<td>4</td>
<td>750,054</td>
<td>0.0019</td>
<td>0.0020</td>
<td>86.0763</td>
<td>0.0020</td>
<td>84.5519</td>
</tr>
<tr>
<td>5</td>
<td>1,080,054</td>
<td>0.0018</td>
<td>0.0019</td>
<td>87.7733</td>
<td>0.0019</td>
<td>86.6336</td>
</tr>
<tr>
<td>6</td>
<td>1,470,054</td>
<td>0.0017</td>
<td>0.0018</td>
<td>89.0177</td>
<td>0.0018</td>
<td>88.0994</td>
</tr>
</tbody>
</table>

The table 4.2.1 shows the various size of the image from figure 4.1 (e) for Constant Message, whereas we used the constant message length as 65 and the key as”soon” in our experiment. The values tells us about the system clearly which is secured even for the larger size of image also gives the small amount of MSE (Mean Square Error) values with good quality image as watermark image. By seeing their PSNR values are high, it is the good evidence for the quality of the image. Thus with noise and other edge method, for constant message length as 20 and the key as”se” the values are calculated as follows in below table 4.2.2 respectively.

**Table 4.2.2 Experiments with different size of the image with various methods**

<table>
<thead>
<tr>
<th>Exp. #</th>
<th>Size of Image (in bytes)</th>
<th>Canny Method Without Noise</th>
<th>Canny Method With Noise</th>
<th>Log Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MSE</td>
<td>PSNR</td>
<td>MSE</td>
</tr>
<tr>
<td>1</td>
<td>196,662</td>
<td>0.0028</td>
<td>78.4109</td>
<td>0.0050</td>
</tr>
<tr>
<td>2</td>
<td>270,054</td>
<td>0.0024</td>
<td>80.4649</td>
<td>0.0047</td>
</tr>
<tr>
<td>3</td>
<td>480,054</td>
<td>0.0019</td>
<td>83.9291</td>
<td>0.0046</td>
</tr>
<tr>
<td>4</td>
<td>750,054</td>
<td>0.0018</td>
<td>86.0825</td>
<td>0.0044</td>
</tr>
<tr>
<td>5</td>
<td>1,080,054</td>
<td>0.0017</td>
<td>87.7733</td>
<td>0.0042</td>
</tr>
<tr>
<td>6</td>
<td>1,470,054</td>
<td>0.0017</td>
<td>89.0209</td>
<td>0.0040</td>
</tr>
</tbody>
</table>
The below figure shows the scatter plot of the various size of the image (in bytes) and their PSNR value. From the figure 4.12, you can observe that the PSNR value is low for small size of image and increases gradually for the higher value size of images. Thus shows we can embed message in large images also with safe and secured. Likewise from figure 4.13, the MSE value is high for small size of image and decreases gradually while increasing the size of the image. The proposed system works well with the small and large size of images also with good quality.

Figure 4.12 Plotting values of different size of image (in bytes) and their PSNR values.

Figure 4.13 Plotting values of different size of image (in bytes) and their MSE values.
4.2.3 Tests with various size of message

The resulted experiments for various size of message in the same image (i.e) figure 4.1(d), whereas here image is Constant. Their corresponding PSNR (Peak Signal to Noise Ratio) and MSE (Mean Square Error) values are noted in the below table 4.3 respectively.

Table 4.3 Experiments with different size of the message with constant image.

<table>
<thead>
<tr>
<th>Exp #</th>
<th>Size of Message</th>
<th>Proposed Method (With Noise)</th>
<th>Proposed Method (With Noise)</th>
<th>Two Individual Method (Author)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MSE</td>
<td>PSNR</td>
<td>MSE</td>
</tr>
<tr>
<td>1</td>
<td>250</td>
<td>0.0050</td>
<td>88.0946</td>
<td>0.0052</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>0.0049</td>
<td>88.0961</td>
<td>0.0051</td>
</tr>
<tr>
<td>3</td>
<td>750</td>
<td>0.0048</td>
<td>88.0986</td>
<td>0.0050</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>0.0047</td>
<td>88.1031</td>
<td>0.0049</td>
</tr>
<tr>
<td>5</td>
<td>1500</td>
<td>0.0046</td>
<td>88.1095</td>
<td>0.0048</td>
</tr>
<tr>
<td>6</td>
<td>1750</td>
<td>0.0045</td>
<td>88.1159</td>
<td>0.0047</td>
</tr>
<tr>
<td>7</td>
<td>2000</td>
<td>0.0044</td>
<td>88.1207</td>
<td>0.0046</td>
</tr>
<tr>
<td>8</td>
<td>2250</td>
<td>0.0043</td>
<td>88.1268</td>
<td>0.0045</td>
</tr>
<tr>
<td>9</td>
<td>2500</td>
<td>0.0042</td>
<td>88.1328</td>
<td>0.0044</td>
</tr>
<tr>
<td>10</td>
<td>2708</td>
<td>0.0041</td>
<td>88.1352</td>
<td>0.0043</td>
</tr>
</tbody>
</table>

The table 4.3 shows the various size of the message of the original image 4.1(d) Image for Various Message, whereas we used the constant image in our experiment. The values tells us about the system clearly which is secured even for the large size of message also gives the small amount of MSE (Mean Square Error) values when there is enough space of edge detection. So that it gives good quality image as watermark image by seeing their PSNR values up to the message length as 2708 in our case.

As the image 4.1 (d), has enough edge pixel values to embed the message length up to 2708, if the message length increases more than 2708 then in that case we can’t get the decrypt message and it shows that edges are important in the proposed method. Moreover, in that case decryption algorithm doesn’t work proper and we can’t get the decrypt message if there is not enough edge to embed the message. The below figure shows the scatter plot of the various size of the image (in bytes) and their PSNR value as evidence. From the figure 4.14, you can observe that the PSNR value increases gradually for the higher value size of messages up to the level of 2708, but it gets stopped in 1750 message long in the previous work because the images do not have sufficient ”edges” to accommodate (embed) the message long of 2000. So compare to the previous method, we can embed more number of message as capacity is more in our proposed method.

Likewise, the MSE value decreases gradually while increasing the size of the message up to the level of 2708, but it stops there when the size of the message goes above 2708 because the images do not have sufficient edges to accommodate (embed) the message.
As shown in the below figure 4.14, the PSNR and MSE value gets stops in some extend so if it gets stopped in their values then it shows that there is no “enough edges” in the image pixel which is to embed the message. So the proposed watermark technique depends on the edges of the image pixel respectively.

Figure 4.14 Plotting values of different size of message (in letters) and their PSNR and MSE values.
4.2.4 Capacity of the message in an Image

In the implementation part, we are going to see how much text can be embedded in the image. As in our proposed algorithm, capacity of the message will depend upon the amount of edges in an image. After the implementation process by seeing the above experimented table 4.1 and 4.2, we conclude that for embedding message in the proposed watermarking scheme, size does not matter as the “edges” are important. To prove edges are important and size of the image are not important, I have shown edges of the figures 4.1(c) and 4.1(b) of small size image (220*229) with more edges and large size image (266*190) with less edges respectively.

While considering these two images, in small image we have more number of edges, so it is possible to encrypt 605 (alphabets) long message. But in the large image we have less number of edges, so it is possible to encrypt 403 (alphabets) long message. So it proves, that size of the image doesn’t matter but edges are important, when need in embedding large amount of messages for the proposed encryption algorithm. To find the capacity of the message, just find the edges in that image and divide by 8 as follows

\[
\text{Number of message can be embed in the image} = \frac{\text{Total number of edges in the image}}{8}
\]

As far the canny edge detector is used in the image we can get better result for hiding them in their edge pixel of the Least Significant Bit (LSB) as Steganography Technique and addition of cryptographic algorithm to make the system robust and reliability. Even though we use canny edge detector, the edges present in the image is contributing for embedding the message up to the availability of the edges in that image.
4.2.5 How strong is the encryption based on different cases?

To show the (robust/strongest) status of the encryption for all the combination by seeing their worst case and best case among all the major three factors such as encryption key size, message size and image size. So for worst case I have named “low” and the best case as “high” respectively.

Size of all 3 factors such as

Key ➔ Low = 4 as “rain” and High= 12 as “SwedenMSe11y”

Message ➔ Low = 20 and High= 300.

Image ➔ Low= 256 bytes and High= 1024 bytes.

By seeing the below table and figures, we can conclude in two aspects as low image (256) and for high image (1024). When talking about low and high image means it indicates the size of the image. For low image, experiments case such as 1, 3, 5 & 7 and for high image experiments case such as 2, 4, 6 & 8 respectively.

For low image, the experiment 5 [ Key(high) , Message(low) ] is “best” and the experiment 1 [ Key(low) , Message(low) ] is “good” and the other two cases 3 & 7 are “worst” due to lots of message as high in both the cases. As it is for low image, we can’t embed more number of messages in the small size of image, if so also it doesn’t decrypt the correct message. So for low image, we should not choose high message.

For high image, the experiment 8 [ Key(high) , Message(high) ] is “best” and the experiment 4 [ Key(low) , Message(high) ] and 6 [ Key(high) , Message(low) ] is “better”, finally experiment 2 [ Key(low) , Message(low) ] is good.

From the below figure 4.16, we can see which factors is best for both low image and high image. As for figure, y-axis takes place with the MSE and PSNR values. For x-axis, we have the combination of both (Key, Message) for low case the value has mentioned as “0” and high case value as “1” respectively.
Table 4.4 Experiments for showing which case is best for strong encryption.

<table>
<thead>
<tr>
<th>Exp #</th>
<th>Key</th>
<th>Message</th>
<th>Image</th>
<th>MSE</th>
<th>PSNR</th>
<th>Rankings based on PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>0.0035</td>
<td>76.3970</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>0.0050</td>
<td>88.0953</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>0.0036</td>
<td>76.2194</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>0.0049</td>
<td>88.0961</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>0.0034</td>
<td>76.3978</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>0.0049</td>
<td>88.0960</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>0.0036</td>
<td>76.2485</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>0.0048</td>
<td>88.0972</td>
<td>1</td>
</tr>
</tbody>
</table>

We can observe from the below figure 4.16, that for high image the MSE and PSNR value graph is smooth and works proper. But for low image, both MSE and PSNR value graph is not perfect, as the problem is “with high message in the low image” for clear view we can see the above table 4.4 value for experiment 3 & 7. As said earlier for low image, we can’t embed more number of messages in the small size of image, if so also it doesn’t decrypt the correct message. So for low image, we should not choose high message respectively.
4.3 Discussion

We noted that the system was success to satisfy many goals that we can conclude them in the following points:

First, the recorded PSNR from different experiments shows that the system successes to hide a message in the Stego-image without appear notable changes in the Stego-image. The system takes the advantage of human visual system which cannot recognize little changes in some pixels of the image. This is the main goal of any steganography system.

Second, the using of the Extension of Vigenere Square algorithm in the system is producing the effect of the two main operations in any cryptography system, which is the substitution operation. This is done by the algorithm through mapping a character from the message to more than one value of the pixel. Also the Extension of Vigenere Square table helps the system to use the second operation of any cryptography system, which the key can be combination of both numbers and text. This can be done by rotating the sequences of characters in the Extension of Vigenere Square table to produce many others substitution values for each characters in the message. This can be done at each time that the algorithm takes a next character from the message. Therefore, it is truly that the system works as a cryptography system in addition to its research as a steganography system even if this is done as in simple way.

Figure 4.16 MSE and PSNR values for low image and high image w.r.t (Key, Message).
4.4 Limitations

The knowledge of the technology is still limited to mainly the research individuals and academia; however there is a growing understanding that this LSB technology is used widely due to its simplicity. But we have a hint in that technique by not taking whole image pixel, as far it takes only the edges of the image for hiding the text. We have limit in the size of the message; when the image has less or poor edges, so we need to take more edge image for this watermark technique. In future, we would extend the system to be more robust and efficient. The research will include the enhancement of the algorithm that will utilize the entire image for embedding the message. We will also analyze the processing time for new introduce method(s) to minimize the time. The major limitation of the application is designed for bit map images (.bmp). It accepts only bit map images as a carrier file, and the compression depends on the document size as well as the carrier image size.
Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In the present world, the data transfers using internet is rapidly growing because it is so easier as well as faster to transfer the data to destination. So, many individuals and business people use to transfer business documents, important information using internet. Security is an important issue while transferring the data using internet because any unauthorized individual can hack the data and make it useless or obtain information unintended to him.

The proposed approach in this project uses a new steganographic approach called image steganography. The application creates a stego image in which the personal data is embedded and is protected with a password which is highly secured. The main intention of the project is to develop a steganographic application that provides good security. The proposed approach provides higher security and can protect the message from stego attacks. The image resolution doesn’t change much and is negligible when we embed the message into the image and the image is protected with the personal password. So, it is not possible to damage the data by unauthorized personnel.

The designed image steganography system is using the bits of pixel in the Stego-image at their edges to hide the characters of the message. The proposed approach in this project uses a new steganographic approach called image steganography. The application creates a stego image in which the personal data is embedded and is protected with a password which is highly secured. After testing the system and studied the recorded results from the experiments, we recommend using this proposed system in hiding secure information in any digital system, because this system collect the properties of both steganography and cryptography sciences.

I used the Least Significant Bit algorithm in this project for developing the application which is faster and reliable and compression ratio is moderate compared to other algorithms. By using the Extension of Vigenere Square, we can add the numbers also in the key such that the password can be more secure with the combination of the characters and numbers. So the main advantage of secure system is password, when password has the combination of both characters and numbers then it is more secure and advantage for the proposed image steganography. Moreover LSB techniques can be used widely but we have used some hints there by taking only the edge pixel in the image. So if the unauthorized person knows the LSB technique also, he doesn’t know the hint as edge in our proposed watermarking system.
5.2 Future Work

The future work on this project is to improve the compression ratio of the image to the text. This project can be extended to a level such that it can be used for the different types of image formats like .bmp, .jpeg, .tiff etc., in the future. The security using Least Significant Bit Algorithm is good but we can improve the level to a certain extent by varying the carriers as well as using different keys for encryption and decryption.

We can experiment this watermarking technique in the frequency domain of various watermark application, whereas watermark as image itself. Also we can implement in other spatial domain techniques and cryptography algorithms for most advanced encryption technique to encrypt the messages.
Glossary

Cover image - An image containing an embedded message.

Cipher text – Refers to encrypted data.

Cryptography – The art of protecting information by encrypting it into an unreadable format, called cipher text. A secret key is used to decrypt the message into plain text.

Encryption – The translation of data into a secret code.

Least significant bit (LSB) - The bit contributing the least value in a string of bits.

Plain text -- Refers to any message that is not encrypted - also called clear text.

S_key – Produced key while encrypting the message, said to be Stego Key.

C_key – Crypto key which changes the plain text into cipher text.

Steganalysis -- The art of discovering and rendering useless covert messages.

Steganography -- A means of overlaying one set of information ("message") on another (a cover).

Stego text -- It is the result of applying some steganographic process to a plain text (not necessarily encrypted).

Mean Square Error (MSE) -- which will be in the range of zero, while two images are identical

Peak Signal to Noise Ratio (PSNR) -- Ratio between the maximum possible power of a signal and the power of corrupting noise in the image.
References


[24] Vigenere Square Table [Online] (Updated 27 April 2011) Available at: <http://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher>


