

Modifying of Hill Cipher using Randomized Approach

Thakwan Akram Jawad

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February, 2014
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Isik Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Ersun Iscioglu
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Ersun Iscioglu
2. Asst. Prof. Dr. Gürcü Öz
3. Asst. Prof. Dr. Ahmet Ünveren

ABSTRACT

In the modern technological communication networks, where data or image transfer through the communication devices has acquired significance security of the data or image transfer has become a major issue. By means of cryptography, they convert plaintext into ciphertext, or into data that must be unreadable to the attacker. Another approach is to encrypt data of image and then send it to the receiver. Various approaches of modifying Hill Cipher have been discovered earlier, which includes advanced and complicated strategies to overcome the problems faced by using Hill image cipher. This thesis is concerned with a rather simple modification on Hill Cipher using randomized permutation of pixel locations approach and comparing the results with Hill image cipher.

The Hill Cipher has advantages in encryption of symmetric data. However, it is insufficient to known plaintext attack. For this reason, the plain image should be processed first to hide its pattern that has the main information. Many researchers used complicated approaches that sometimes depend on matrix mathematics and sometimes on information manipulations. The aim of these complication strategies is to make the encryption approaches in thwart order to any known plaintext and ciphertext attacks. The main object of this thesis is to adjust the original Hill Cipher to decrease the susceptibility to known plaintext attacks and known ciphertext attack. This is achieved by randomizing the information locations for the whole image. So, even if the attacker can guess the key, the decryption still gives false information. Both of Hill image cipher and the proposed modification on it (termed MHill in this thesis) are convenient for all images plaintext block encryption. Results from

statistical analysis and comparative studies have shown that Hill Cipher with randomized approach has a correlation coefficient value which is closed to zero and has the maximum deviation value. According to these results Hill Cipher with randomized approach proved that it has better encryption quality compared to Hill image cipher.

Keywords: Hill Cipher, randomized approach, modified Hill image cipher, encryption, decryption, Hill Cipher with randomized approach.

ÖZ

Modern teknolojik iletişim ağlarında, iletişim araçları vasıtasıyla gerçekleştirilen veri ve resim transferlerinde önemli güvenlik sorunları olmuştur. Kriptografi ile düz bir metin kodlanmış bir hale getirilebilir ya da bir veri saldırganlara karşı okunamaz hale dönüştürülebilir. Bir diğer korunma yöntemi de bir resmin verisini kodlama ve göndermedir. Bu çalışmada, Hill Cipher kullanılarak basit bir şekilde pixel yerlerinde rastgele değişimler yapılmış ve elde edilen sonuçlar standart Hill Cipher sonuçları ile karşılaştırılmıştır.

Hill kodlama yöntemi simetrik verilerin kodlanmasında bir çok avantaj sağlamaktadır. Fakat bu yöntem bilinen saldırılara karşı yetersizdir. Bu nedenle resim öncelikle gizli bir örüntü ile işlenmelidir. Birçok araştırmacı bazen matris matematiği, bazen de manipule edilmiş bilgiler üzerinde karışık kodlamalar kullanmaktadır. Bu stratejilerin amacı bilinen düz metin ya da şifreli metin saldırılarını önlemek için kodlama yapmaktır. Bu tezin amacı da, bilinen düz metin ve şifreli metin saldırılarına karşı duyarlılığı azaltmak için yeni bir Hill Cipher yapılandırmaktır. Bu yeni yapılandırmada, görüntü içerisindeki tüm piksellerin yerleri rastgele değiştirilir. Böylece saldırgan şifreyi tahmin edebilse bile, çözümleme de yanlış bilgi elde edecektir. Hill Cipher ve yeniden tasarlanmış hali (MHill Cipher) tüm şifresiz resimler için blok kodlama kullanmaktadır. İstatiksel analiz ve karşılaştırma çalışmalarının sonuçları göstermektedir ki, MHill Cipher'ın korelasyon katsayısı sıfıra yakındır ve yüksek sapma değerindedir. Bu sonuçlara göre bu çalışmada üretilen MHill Cipher'ın kodlama kalitesinin orjinal Hill Cipher'a göre daha iyi olduğu belirlenmiştir.

Anahtar Kelimeler: Hill Cipher, RastlantısalYaklaşım, modifiye edilmiş Hill Cipher, Kodlama, Kod çözme, Rastlantısal Yaklaşımına göre Hill Cipher.

ACKNOWLEDGMENT

I would never have been able to complete this dissertation without the help of the people who have supported me with their best wishes.

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Ersun Iscioglu for his efforts and supports for doing this research. I sincerely thank to the committee members of my thesis defense jury for their helpful comments on this thesis. I gratefully acknowledge Assoc. Prof. Dr. Alexander Chefranov for his helpful notes and guidance during my work in this thesis. I would like to fully appreciate my brother and friend Dr. Firas Hanna Zawaideh for his helpful feedback and for providing useful references. Last but not least I would also like to thank my wife and my children for giving me the courage to leave them in my country while I was studying in Cyprus.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	v
ACKNOWLEDGMENT	vii
LIST OF FIGURES	xi
LIST OF TABLES	xiii
LIST OF ABBREVIATIONS	xiv
1 INTRODUCTION	1
1.2 Research Objectives	3
1.3 The Quality of a Cipher System	4
1.4 Contributions	4
1.5 Thesis Layout	5
2 LITERATURE REVIEW.....	6
2.1 The Hill Cipher.....	6
2.2 Interlacing and Decomposition	8
2.3 Two-Stage Hill Cipher technique	11
2.4 Advanced Hill Cipher.....	13
2.5 AdvHill Algorithm Setback.....	17
3 THE PROPOSED MODIFICATION	19
3.1 Introduction	19
3.2 The Proposed Modified Hill Cipher (MHill)	20
3.2.1 Illumination Re-quantization	21

3.2.2 Random permutation of pixel locations.....	22
3.2.3 Applying Hill Cipher	25
3.3 The proposed Decipher	28
3.4 The MATLAB Simulation	29
4 RESULTS AND DISCUSSION	32
4.1 Introduction	32
4.2 Ciphering Metrics.....	33
4.2.1 The Correlation Coefficient	33
4.2.2 Similarity	34
4.2.3 Histogram Uniformity	34
4.2.4 The Irregular Deviation Factor	34
4.2.5 Diffusion Metrics	35
4.2.6 Execution Time.....	36
4.3 Testing the MHill Algorithm.....	37
5 CONCLUSION	49
REFERENCES.....	51
APPENDICES	56
Appendix A: The proposed MHill MATLAB Simulation Program	57
Appendix B: The Proposed MHill MATLAB Simulation Program for Highly detailed images only (No De-quantization operation).....	60
Appendix C: The Hill MATLAB Simulation Program.....	62

Appendix D: The Designed Functions that are used in programs in appendices A, B and C	63
Appendix E: The Designed Programs to Measure NPCR and UACI Metrics	66

LIST OF FIGURES

Figure 1.1: The Symmetric cipher system structure	3
Figure 2.1: The flow of encryption and decryption	10
Figure 2.2 (a): The plain image with its histogram using modified Hill Cipher with 11 involutory Key	11
(b): The ciphered image with its histogram using modified Hill Cipher with involutory Key	11
Figure 2.3: Two-Stages Hill Cipher	12
Figure 2.4: The best results of the two-stage algorithm	12
Figure 2.5: Bad encryption based on two-stage Hill Ciphered	13
Figure 2.6: AdvHill Algorithm	15
Figure 2.7: Original images (a-e), corresponding encrypted images by Hill Cipher (f-j) and by AdvHill Cipher Algorithm (k-o)	16
Figure 2.8: Cipher and decipher of detailed scene using AdvHill [10]	17
Figure 2.9: Cipher and decipher of less detailed scene using AdvHill [10]	18
Figure 3.1: Applied Hill Cipher	19
Figure 3.2: Traditional Hill Cipher and MHill Cipher Algorithms.....	21
Figure 3.3: Re-quantization process.....	22
Figure 3.4: Random permutation effect	23
Figure 3.5: The vertical scanning used to change the image dimensions.	26
Figure 3.6: The proposed decipher process of the MHill algorithm	29
Figure 3.7: The image selection window used by the simulation	30
Figure 3.8: The result window of the simulation	31
Figure 4.1: Hill Cipher Histogram for the sample lena.jpg	40

Figure 4.2: MHill Cipher Histogram for the sample lena.jpg	40
Figure 4.3: Hill Cipher Histogram for the sample SP.jpg.....	41
Figure 4.4: MHill Cipher Histogram for the sample SP.jpg	42
Figure 4.5: Hill Cipher Histogram for the sample manara.jpg	43
Figure 4.6: MHill Cipher Histogram for the sample manara.jpg.....	43
Figure 4.7: Hill Cipher Histogram for the sample donald.bmp	44
Figure 4.8: MHill Cipher Histogram for the sample donald.bmp.....	44
Figure 4.9: Hill Cipher Histogram for the sample image1.bmp	45
Figure 4.10: MHill Cipher Histogram for the sample image1.bmp.....	46
Figure 4.11: Hill Cipher Histogram for the sample image2.bmp	47
Figure 4.12: MHill Cipher Histogram for the sample image2.bmp.....	47

LIST OF TABLES

Table 3.1: Permutation map for rows of the image.....	22
Table 3.2: Permutation map for columns of th.....	23
Table 3.3: Final Results for the MHill Algorithm.....	27
Table 4.1: MHill and Hill Quality Tests.....	37
Table 4.2: Results obtained by proposed MHill and Two-Stage Hill for Lena.jpg	
Image	48

LIST OF ABBREVIATIONS

AdvHill	Advanced Hill
ASCII	American Standard Code for Information Interchange
bmp	Bitmap
Cc	Correlation Coefficient
dB	Decibels
IDF	Irregular Deviation Factor
jpg	Joint Photographic Group
MHill	Modified Hill
MSE	Mean Square Error
NPCR	Number of Pixels Change Rate
RGB	Red Green Blue
UACI	Unified Average Change Intensity

Chapter 1

INTRODUCTION

Regarding to the advance in network expertise, data security is a progressively significant difficulty. Many applications of multimedia expertise and progressively transmission capacity of mesh gradually direct us to acquire information exactly and apparently through images. Cryptography, the research of encryption, plays a centered function in wireless telephone communications, e-commerce, pay-tv, sending personal emails, conveying economic data, computer passwords, security of ATM cards and feels on numerous aspects of our everyday inhabits [1]. Cryptography is the art or research including the values and procedures of changing an obvious note (plaintext) into one that is non-obvious (ciphertext) and then convert that message back to its initial pattern. In modern times, cryptography is considered to be agency of both numbers and computer science and it is related nearly with data idea, computer security, and technology.

1.1 The Cipher System Overview

Substitution cipher is one of the basic constituents of classical ciphers. A substitution cipher is a method of encryption by which units of plaintext are exchanged with ciphertext according to a normal system; the flats may be lone notes, in twos of notes, triplets of notes, blends of the overhead, and so forward. The receiver deciphers the text by accomplishing an inverse substitution. The units of the plaintext are retained in the identical sequence as in the ciphertext, but the flats themselves are

changed. There are a number of different types of substitution cipher. If the cipher operates on single notes, it is termed a simple substitution cipher; a cipher that functions on bigger groups of notes is termed poly-graphic. A mono-alphabetic cipher uses repeated substitution over the whole note, while a poly-alphabetic cipher utilizes a number of substitutions at different times in the note—such as with homophones, where a unit from the plaintext is mapped to one of some possibilities in the cipher text. Hill Cipher is a kind of mono-alphabetic poly-graphic substitution cipher which is simply can be known as a block cipher that has some benefits such as hiding note intensity of the plaintext. Hill Cipher can be implemented simply because of matrix multiplication and inversion for the use of enciphering and deciphering [2].

Moreover, encryption schemes were the first centered locality of interest in cryptography. They deal with supplying means to endow private communication over an insecure channel. A sender wishes to transmit data to a receiver over an insecure channel that is a conduit which may be tapped by an adversary. Thus, the data to be broadcast, which we call the plaintext, should be transformed (encrypted) to a ciphertext, a form not legible by anybody other than the proposed receiver. The last mentioned should be granted some way to decrypt the ciphertext, retrieve the original message, while this must not be likely for an adversary [2].

This is where keys arrive into play; the receiver is advised to have a key at his disposal, endowing him to retrieve the actual note, a detail that distinguishes him from any adversary. The encryption algorithm changes plaintexts into ciphertexts while the decryption algorithm converts ciphertexts back into plaintexts. The

following flowchart in Figure 1.1 below shows the basic symmetrical cryptosystem structural design that is used for image cipher [2].

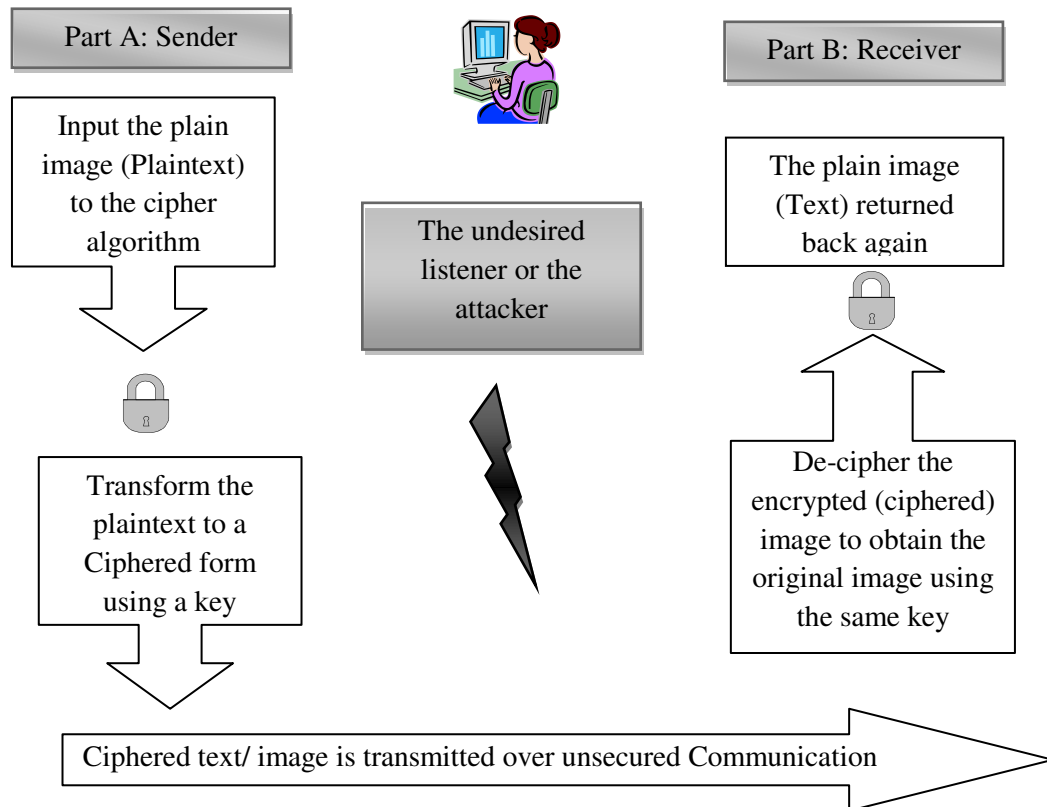


Figure 1.1: The Symmetric cipher system structure

1.2 Research Objectives

Hill Cipher is the first poly-graphic cipher where the plaintext is split up into assemblies of contiguous letters of the same steady extent m , each group is then changed into a distinct assembly of m letters. This poly-graphic feature expanded the throughput and speed of Hill Cipher. Also, other benefits may exist from poly-graphic cipher in facts and figures encryption such as its resistance to frequency implementation [3].

Unfortunately, the Hill Cipher did not work efficiently for ciphering images. This fact came from the reality that images sometimes have many repetitive values if it

contains low details [4]. Those values are zeros for any probable black object may appear in the scene while usually the backgrounds are white in such cases. So, details in those cases will remain and hence the main objective of ciphering goes in vain.

It is intended in this work to use the Hill Cipher with modifications that lead to proper cipher system making a full use of the advantages of the Hill Cipher security.

1.3 The Quality of a Cipher System

The last step of designing a cipher system is to measure its quality. This measurement may be based on visual inspection or using quantitative metrics (i.e. correlation coefficient, similarity, histogram uniformity and irregular deviation factor). The human eye can determine if the image cipher system removed the unique pattern that identify the scene in the picture, although; this subjective test is not enough to compare between different systems [5]. Therefore, it is intended to use many different quality metrics in this research. In fact, those metrics, side by side with visual inspection, paved the way to choose the better algorithm from many tried strategies.

1.4 Contributions

This thesis is highly related to [1], [2] and [3] references. We aimed from our proposed algorithm is to find out:

- High degree of security against the attacker by the use of permutation.
- High efficient algorithm for image encryption with the large single color area.
- Very low correlation coefficient between plain and ciphered image.
- Very influential of one-pixel change on the whole image.

1.5 Thesis Layout

In addition to what was proposed in Chapter one in this thesis, the next chapters are organized as follows:

Chapter two: Presents many different related studies about image cipher.

Chapter three: It is dedicated for the proposed algorithm (The Modified Hill "MHill") that was designed and tested intensively.

Chapter four: Presents the results of the metrics taken for both the proposed (MHill) and the conventional Hill image cipher. Those results are discussed and compared with giving more details on the meaning of metric used in this research.

Chapter five: Presents the conclusion briefly.

Chapter 2

LITERATURE REVIEW

As it has been discussed in Chapter one, the Hill Cipher cannot encrypt the image perfectly. However, the Hill Cipher is used after conditioning it to suite the image characteristics. The powerful attack immunity aspect of the Hill Cipher made the researchers keen to adopt it for image ciphering system. In this chapter, initially, Hill Cipher operations and mechanism are discussed in details. In the following sections some examples of the attempts to modify Hill Cipher to make it more efficient in image ciphering will be investigated. Those related works are categorized according to their methodologies. Finally, problems found while trying to adopt the famous AdvHill strategy is concerned in a separate section.

2.1 The Hill Cipher

This cipher technique was designed by the mathematician Lester Hill in 1929. Hill Cipher is the first poly-graphic cipher where the plaintext is split up into assemblies of contiguous letters of the same steady extent m , each group is then changed into a distinct assembly of m letters. This poly-graphic feature expanded the throughput and speed of Hill Cipher. Also, other benefits may exist from poly-graphic cipher in facts and figures encryption such as its resistance to frequency implementation [3].

Historically, the Hill Cipher was designed to cipher the written text, so the following discussion is dealing with text.

The core of Hill Cipher is matrix manipulation. For encryption, algorithm takes m successive plaintext letters and rather than of that substitutes m cipher letters. Its linear algebra formula is $C = K \times P \pmod{m}$, where C comprises the ciphertext block, P comprises the plaintext impede and K is the key. Whereas, in decryption side, a key inverse matrix (K^{-1}) is required [5].

In Hill Cipher, each character is assigned a numerical value that is starting with 0 (zero). The numerical value is illustrated like $a=0, b=1, \dots, z=25$ [5]. The substitution of ciphertext letters in the location of plaintext letters directs to m linear equation. For $m=3$, the system can be described as follows [9]:

$$\begin{aligned} C_1 &= (K_{11}P_1 + K_{12}P_2 + K_{13}P_3) \pmod{26} \\ C_2 &= (K_{21}P_1 + K_{22}P_2 + K_{23}P_3) \pmod{26} \\ C_3 &= (K_{31}P_1 + K_{32}P_2 + K_{33}P_3) \pmod{26} \end{aligned} \quad (2.1)$$

This case can be conveyed in terms of column vectors and matrices, and can be described as follows [9]:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \quad (2.2)$$

We can compose easily as $C = KP \pmod{26}$, where C and P represent the plaintext and ciphertext respectively as a pillar vectors of extent 3, and K is a 3×3 matrix, which has to be encrypted from the attack of opponent. In the decryption side an inverse key matrix (K^{-1}) is applied.

The inverse key matrix (K^{-1}) can be satisfied by the equation of $KK^{-1} = K^{-1}K = I$.

While, it is not necessary that the inverse matrix of the key always satisfy. K^{-1} is

applied to the ciphertext, and then the plaintext is recovered. Generally it can be summarized as follows [9]:

For encryption:

$$C = E_k(P) = K_p \quad (2.3)$$

For decryption:

$$P = D_k(C) = K^{-1}C = K^{-1}K_p \quad (2.4)$$

Where I is the identity matrix and each of E_k and D_k represents the Encrypted and Decrypted Key respectively

2.2 Interlacing and Decomposition

Sastry and Shankar, proposed a modified Hill Cipher for interlacing and iterations [7]. Although, the use of interlacing iterations in ciphering and decomposition iterations for encryption / decryption strategy generates a more complicated steps but it leads to have a more secure algorithm. In interlacing technique, the positions of bits of the plaintext that are presented in the matrix will be permuted. They designed the interlacing and decomposition by implementing the following steps [7]:

1. Change the elements of matrix into binary form of 8 bits.
2. The new dimension now of the matrix is $n \times 8n$. Split the generated matrix into two matrices with dimension $n \times 4n$.
3. Swap $(2m)^{th}$ column of first matrix with $(2m-1)^{th}$ column of the second matrix with m varies from 1 to $2n$.
4. Combine the two matrices to create a new $n \times 8n$ matrix.
5. Translate each 8 bit elements to its equivalent decimal numbers to get $n \times n$ matrix.
6. The interlaced matrix becomes various from the original matrix.

However, using this kind of repetitive iterations may lead to the following problems [8]:

1. If each of the plaintext matrix or either the key matrix or its inverse matrix contains indices with real numbers (fractions), in such case the elements cannot be represented in a binary form then the algorithm will be failed.
2. In decryption side, a key inverse matrix is applied for the multiplication linear algebra formula. But in the case of that the key matrix was not invertible then this algorithm will be failed.

Acharya *et al.* had proposed a solution to the above problems in image ciphering using interlacing and decomposition. They proposed that all the elements should be pure integers and a self-reversible matrix with integer elements [8].

The suggested algorithm to generate that self-invertible matrix, or what is termed in reference [8] by an Involutory matrix, was as follows [8]:

1. A random non-singular matrix A_{22} of dimension $m/2 \times m/2$ with integer elements is generated.
2. Now $A_{11} = -A_{22}$
3. A random number k ($k > 1$) is selected.

Now

$$A_{21} = (I - A_{11}) / k \quad (2.5)$$

$$A_{12} = (I + A_{11}) \times k \quad (2.6)$$

4. The generated involutory key matrix K will be formed as follows:

$$K = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (2.7)$$

The researchers showed that this involutory key matrix can be verified with any size regards to the dimensions of plain image. Furthermore, this key is involutory matrix which means that $K^{-1} = K$. So the same matrix can be used for both ciphering as well

as deciphering [8]. Finally, the flow diagram of encryption and decryption algorithm according to this research is shown in Figure 2.1.

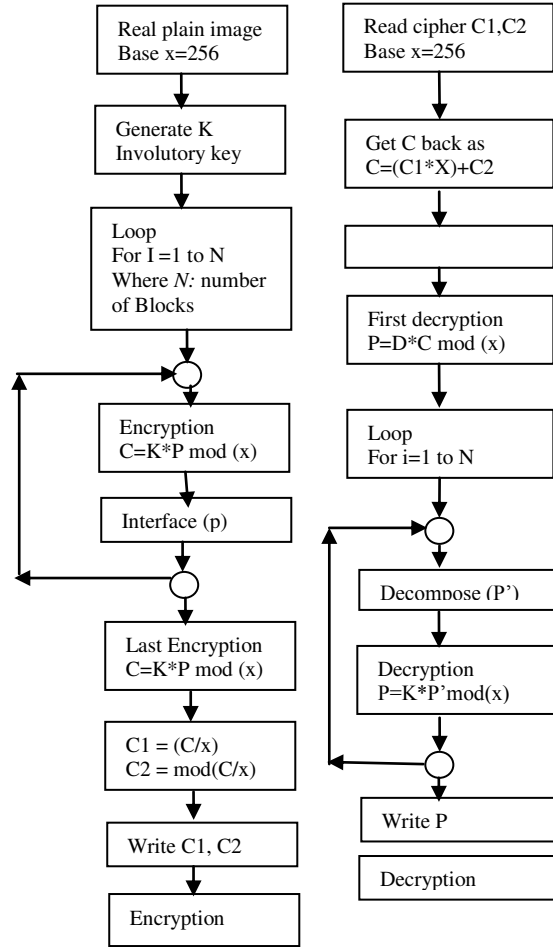


Figure 2.1: The flow of encryption and decryption [8]

What happened here is converting the plain image into a cipher image through two steps of cipher encryption. While in the decryption side, the ciphered image that is presented by these two matrices are decomposed to reproduce the original image again. According to the researchers, this method designed to be used specifically in saving the biometrics templates to increases the security [8]. The ciphered image with the histogram was as follows in Figure 2.2.

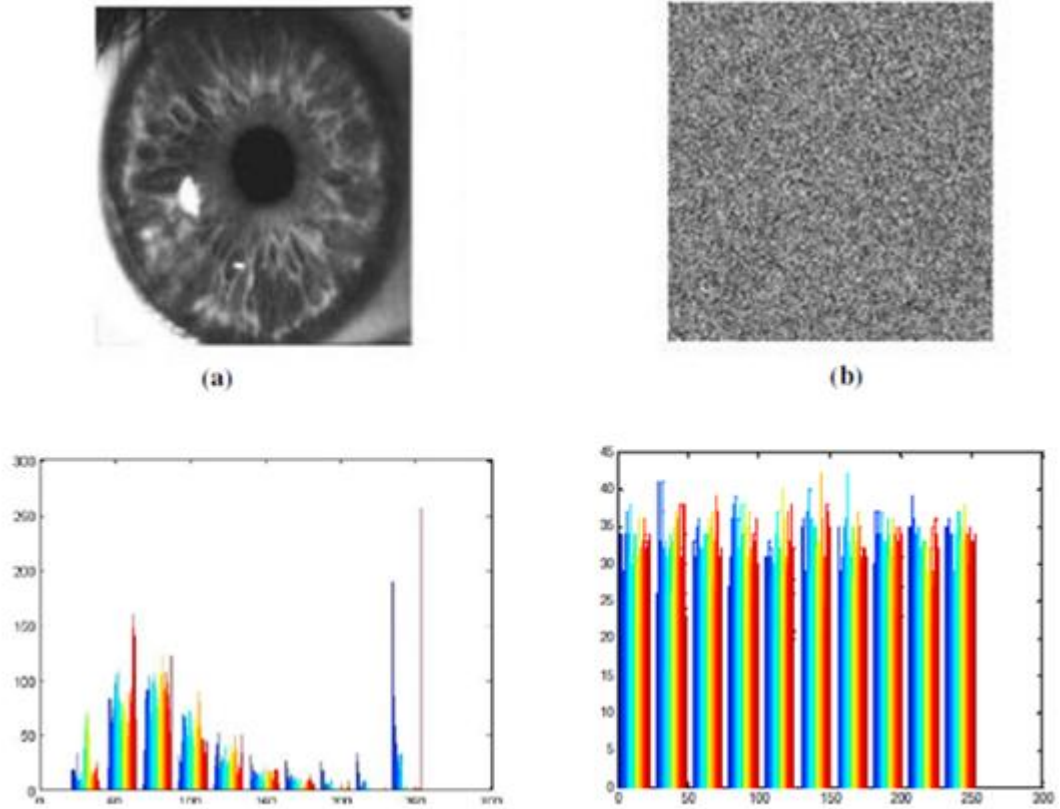


Figure 2.2 (a): The plain image with its histogram using modified Hill Cipher with involutory Key [8].
 (b): The ciphered image with its histogram using modified Hill Cipher with involutory Key [8].

2.3 Two-Stage Hill Cipher technique

Recently, Panduranga H T and Naveen Kumar S K tried another strategy for adapting the Hill Cipher to be suitable for image cipher [9]. They implied that the key should be of minimum eight characters. Later, all keys elements are replaced by predefined 4 out of 8 code vales. Then this key matrix will be converted into a self-invertible key matrix. According to this research, the input image is divided into blocks of 16x16, each block along with self-invertible matrix goes through the Hill Cipher in the first stage to create a ciphered block. Changing the self-invertible matrix is achieved by replacing the diagonal values of the basic block (8x8) by an ASCII value instead of 4 out of 8 code. The Resultant block from the first stage Hill Cipher goes through a second stage Hill Cipher, using the modified key, to obtain the

final ciphered image. As usual, deciphering was proposed to in the reverse of the ciphering process. Figure 2.3 shows the block diagram of the ciphering algorithm proposed in reference [9] by those two researchers.

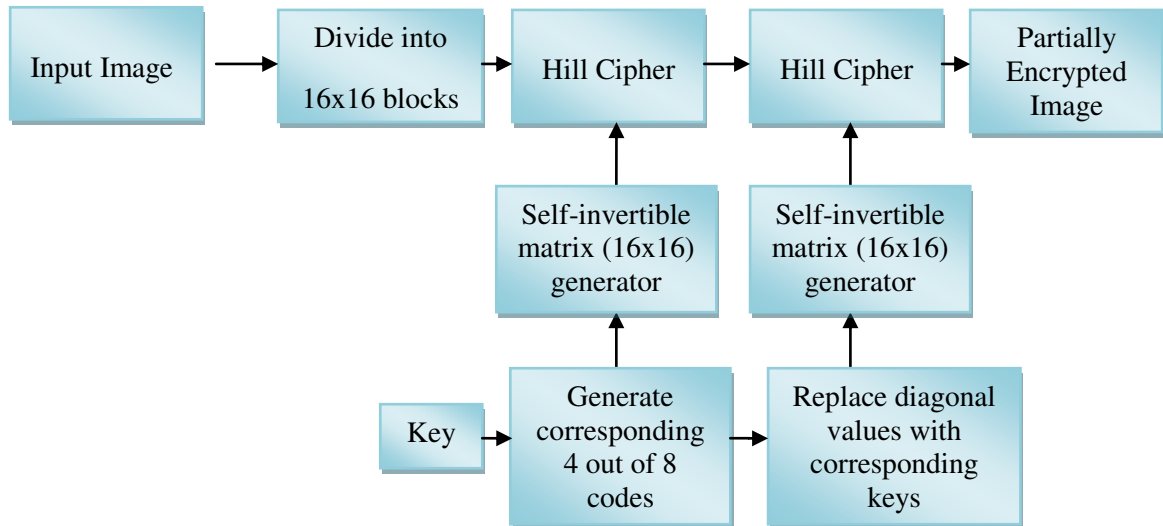


Figure 2.3: Two-Stages Hill Cipher [9].

This research was entitled in "Advanced Partial Image Encryption using Two-Stage Hill Cipher Technique" and was published in 2012 with best results obtain on Lena image example as shown below.

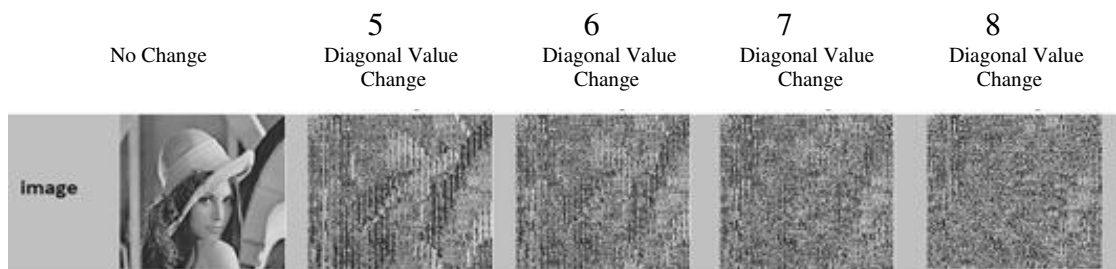


Figure 2.4: The best results of the two-stage algorithm [9]

The above figure shows how the changing of some of the diagonal values in the self-invertible key matrix was effective. The key was used in the first step then few

diagonal values were changed to generate a new self-invertible key matrix that will be used in the second Hill Cipher iteration [9].

However, most of the selected test by this paper was badly ciphered with lower security than the sample above. The security of this cryptosystem was affected because the ciphered image enabled the unwanted inspector to see some details and make him expect what the original image was. This case of low security ciphered image was clearly noticed in the following sample.

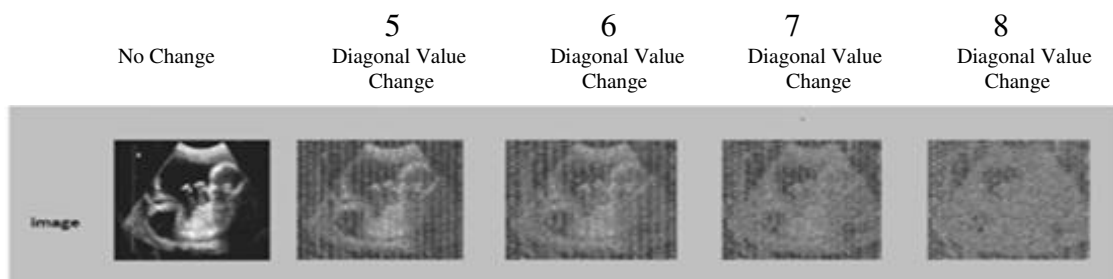


Figure 2.5: Bad encryption based on two-stage Hill Ciphered [9]

According to the researchers who invented this novel two-stage Hill Cipher, this technique could be used as partial image encryption. They claimed that it is enough to secure definite applications like patient information security [9]. This fact is beyond dispute, since the ciphered image in Figure 2.5 removed the detailed information of the patient.

2.4 Advanced Hill Cipher

This algorithm is generated to overcome disadvantages of the other related algorithms. As it was mentioned before, in Hill Cipher encryption algorithm, which used self-invertible key matrix, there are the problem of encryption of image with the existing of same color or pattern in the picture, while in advance Hill Cipher applied an involutory key matrix as encryption the color image can be easily encrypted [10].

Advanced Hill Cipher algorithm follows four main steps for encryption of images:

1. As a first and the main step is that an involutory key-matrix has to be defined with the dimension of m rows and m columns ($m \times m$).
2. The image expected to be encrypted (plain image) should be divided into ($m \times m$) symmetric sub-blocks for encryption.
3. The i^{th} pixels of each sub block should be brought together for constructing the temporary block.
 - a. Next, hill-cipher encryption method is used on the temporary block.
 - b. Then, the output matrix is transposed and again hill-cipher is applied to this transposed matrix.
4. Finally, we obtained the final matrix which is located in the i^{th} block of ciphered image.
5. Steps 3 and 4 are repeated by incremented i till the whole image is scanned.

Those steps are illustrated clearly in Figure 2.6.

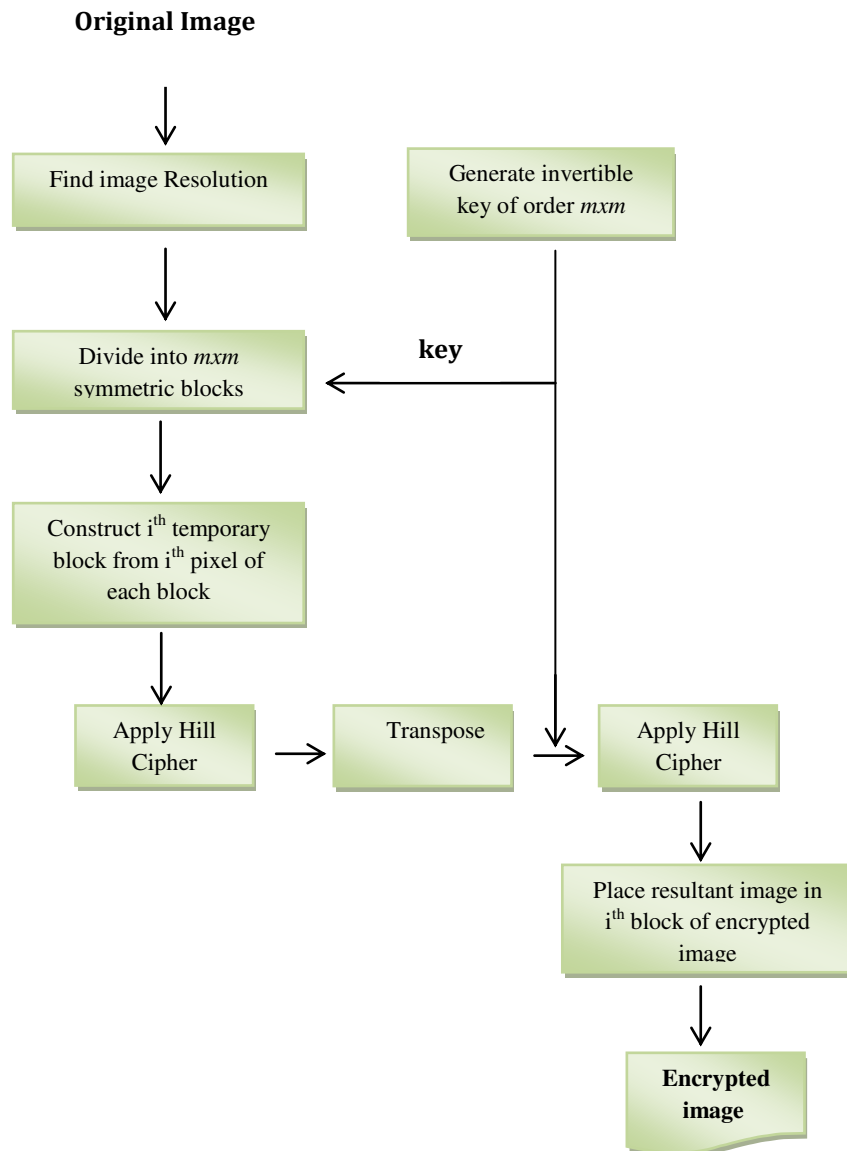


Figure 2.6: AdvHill Algorithm [10]

Researcher who proposed this algorithm did not talk about the decryption. This is because his work was concerned about privacy of biometric traits. The published paper was entitled as "On the Privacy Protection of Biometric Traits: Palm print, Face and Signature" although it showed results of many different scenery images. Figure 2.7 shows those samples as they were stated in that paper.

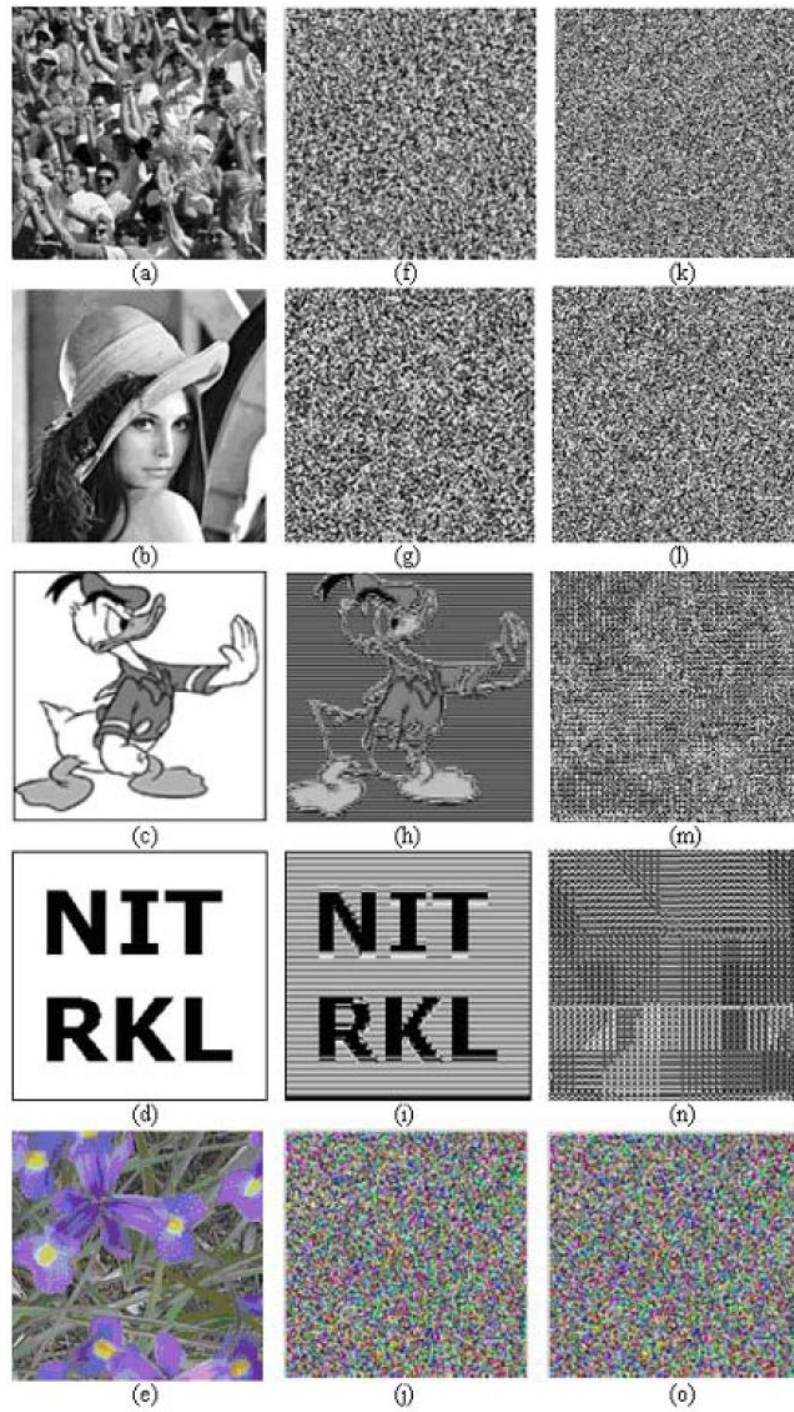


Figure 2.7: Original images (a-e), corresponding encrypted images by Hill Cipher (f-j) and by AdvHill Cipher Algorithm (k-o) [10]

2.5 AdvHill Algorithm Setback

As a matter of fact, decryption is important to analyze the cryptosystem's noise immunity. It can also show the researcher its authenticity in the sense of retrieving back the plain image.

Nevertheless, it is assumed by us that the decryption (deciphering) is the done simply by reversing each step of the AdvHill. However, it has been proved practically through this work that this cipher system did not recover the black and white images. It is proven now that AdvHill works only when there were many details in the image and failed for small objects scenes as shown in Figures 2.8 and 2.9. The following results obtained for the AdvHill deciphering that was experimented through this work [10].

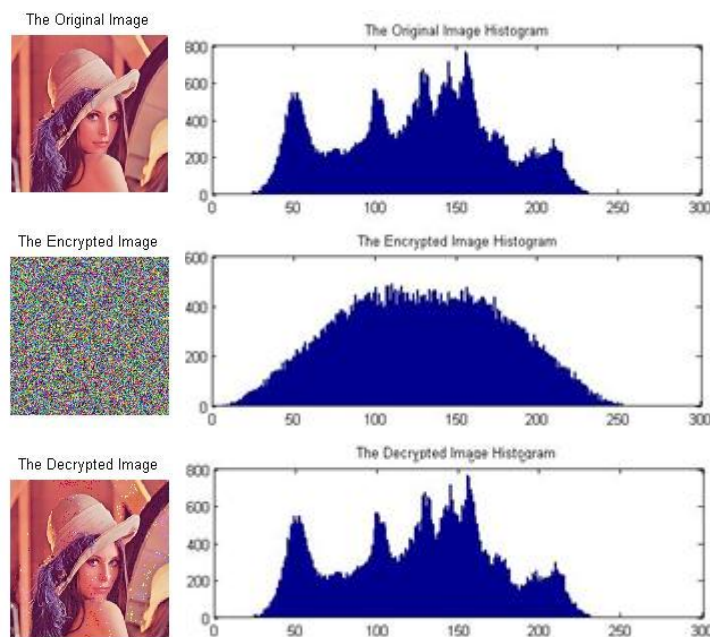


Figure 2.8: Cipher and decipher of detailed scene using AdvHill [10]

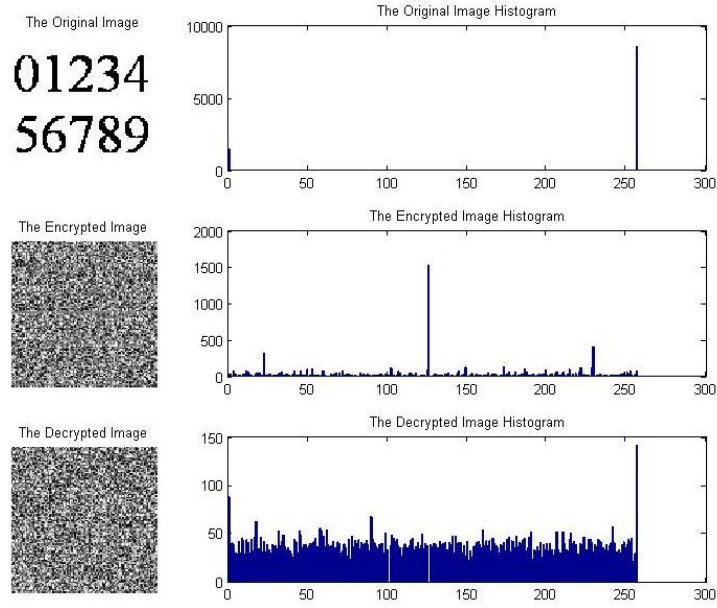


Figure 2.9: Cipher and decipher of less detailed scene using AdvHill [10]

Although AdvHill is a robust and more secure than the traditional Hill algorithm for ciphering images, it failed to return back the images of small objects with wide backgrounds as presented in Figure 2.9.

This problem has been discovered while trying to simulate the AdvHill technique as no other previous paper mentioned this problem. Therefore, this algorithm and the early proposed developments were cancelled later from this work.

Chapter 3

THE PROPOSED MODIFICATION ON HILL CIPHER (MHill)

3.1 Introduction

Since the Hill Cipher is based on matrix multiplication and inverses, it is simply computed and it overcomes the frequency distribution problem of other algorithms used before as illustrated in Figure 3.2.(a). This linearity makes Hill Cipher susceptible even to simple attacks. If an attacker intercepted enough plaintext and ciphertext pairs, a linear system could be set up to calculate the encryption matrix [11]. Another problem associated to the Hill Cipher's linearity is that; it encodes every identical plain image to the same cipher image matrix. These problems arise significantly with images that contain small objects with uniform background as shown in Figure 3.1.

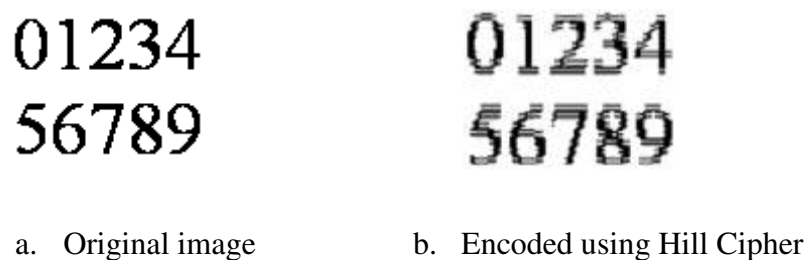


Figure 3.1: Applied Hill Cipher

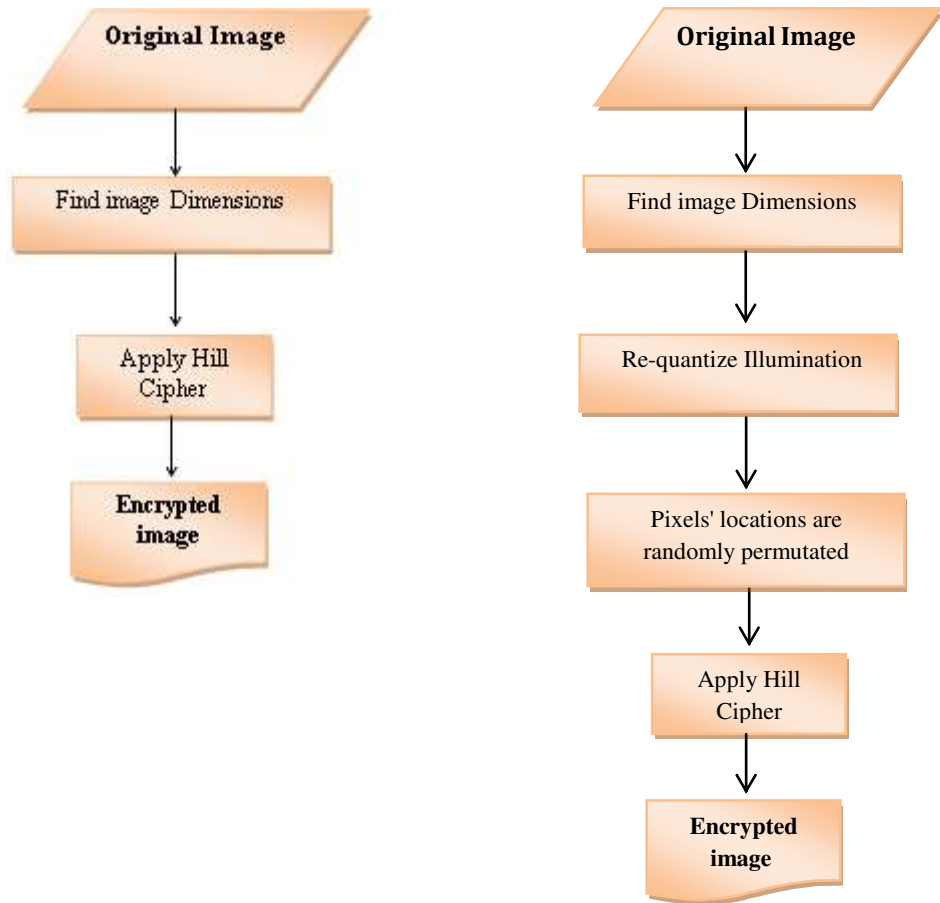
It is obvious that the ciphered image is easily understood by the attacker and hence the encryption goes in vain by the original Hill Cipher. The aim of this work is to overcome this kind of security problems in image ciphering by using Hill technique

preceded through extra steps. As the Hill Cipher is the core of this methodology, this algorithm is termed as Modified Hill Cipher (MHill) for simplicity.

3.2 The Proposed Modified Hill Cipher (MHill)

The proposed algorithm enhances the Hill Cipher which means that it can be used to cipher images efficiently. The basic concept of the modification is to change the illumination quantization and then randomly change the places of pixels before using the Hill Cipher. MATLAB v7.8.0.347 (R2009a) was used to simulate the suggested strategy that is shown in the flowchart in Figure 3.2.(b).

Each step in this work has been used for a certain purpose to promote the Hill Cipher and make it reliable and convenient to be applied on more complicated matrices as illustrated in Figure 3.2.(b). In other words, a suitable Hill Cipher for ciphering the images especially for those of small objects lying on unvarying backgrounds is made. The following sub-sections describe each process of the algorithm.



a. Traditional Hill Cipher

b. MHill Cipher

Figure 3.2: Traditional Hill Cipher and MHill Cipher Algorithms

3.2.1 Illumination Re-quantization

After calculating the image dimensions, illumination re-quantization is applied on the processed image. This adjustment on the intensity is basically used to squeeze the histogram or color map for colored images to remove the zeros from the output of Hill Cipher calculations. It is necessary because usually the plain images of small objects on fixed backgrounds contain many zeros for black objects and many high values for the fixed white background. This arrangement led to many zeros out of the Hill operations (as shown in section 3.1) due to matrices multiplications, then the recovery will give unexpected values.

It is supposed in this work that the intensity resolution is 8 bit, so it varies from zero to 255. This re-quantization is simply making the intensity of zeros to be 1. The resultant quantization will be varying from 1 to 255. There will be no effect seen by the human eye on the image. The following Figure 3.3 shows the illumination re-quantization effect on the image.

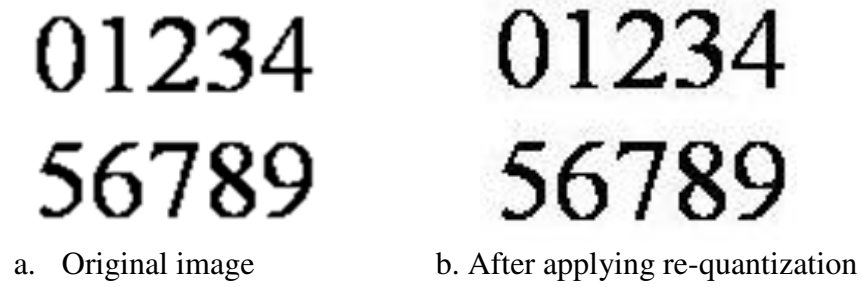


Figure 3.3: Re-quantization process

3.2.2 Random permutation of pixel locations

This is achieved by generating a table that contains integers from 1 to $R_{\max}$, where ($R_{\max}$) represents the maximum row index for the plain image, but they are permuted randomly. Another table for columns is prepared in the same way and it contains random permuted integers from 1 to the maximum column index for the plain image ($C_{\max}$).

These two tables contain the new map for rows and columns separately. Tables 3.1 and 3.2 show an example of a random generating permuted map table for both rows and columns.

Table 3.1: Permutation map for rows of the image

Old Row index	1	2	3	4	5	6	7		$R_{\max}$
New Row index	11	5	22	7	1	16	4		N1

Table 3.2: Permutation map for columns of the image

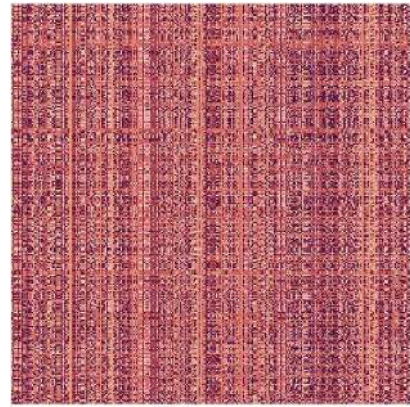
Old column index	1	2	3	4	5	6	7		$C_{\max}$
New column index	19	25	12	30	51	2	28		N2

Where $R_{\max}$ is the maximum row index, $C_{\max}$ is the maximum column index, N1 and N2 are random integers.

Each pixel in each layer of the RGB layers is put in a new location depending on those two tables. For instance, according to the example tables above the pixel in location (1,1) is moved to location (11,19) while the pixel in (1,2) is moved to (11,25). This permutation will increase security by increasing the immunity to known-plaintext attacks. This is because even if the attacker expected the output of the cipher operation, it would be hard for him to generate the permutation tables.



a. Plain image



b. Randomly permuted of image.

Figure 3.4: Random permutation effect

In fact the attacker will obtain this kind of image as shown in (b) for the plain image in (a) in Figure 3.4.

It is true that the "known plain-text" attacker can extract the key used in Hill Cipher by knowing sufficient parts of the plaintext and ciphertext. Surprisingly, by applying the inverse of the key on the ciphered image the output will be as shown above in Figure 3.4 b. This will certainly confuse the attacker and make him think that the supposed "known plain-text" is not known any more.

The random permutation of pixel locations was achieved by MATLAB using the following code:

```
[r c]=size(Image);  
  
for i=1:r  
    for j=1:c  
        img(i,j)=Image(new_row(i),new_colomn(j));  
    end  
end
```

Where (Image) is the input image, (new_row) and (new_colomn) are vectors representing the tables of random permutation for rows and columns respectively.

Although the permutation tables are generated in the sending side, the receiver has to know them as well. Those tables could be altered for each iteration or according to some basis that the transmitter and the receiver agree. Those basis that the sender and receiver can adopt may be time basis or subject basis.

3.2.3 Applying Hill Cipher

The Hill Cipher algorithm is based on linear transformation, and was invented by Lester S. Hill in 1929[13]. Hill Cipher is a block cipher algorithm where plaintext is divided into equal size blocks. The attack on Hill Cipher is tricky as the key is an $n \times n$ matrix which requires too many iterations to guess it. Hill Cipher is no longer used alone due to the vulnerability against known-plaintext attacks [14].

At this step, Hill Cipher is applied using an $n \times n$ key matrix to hide the remaining information. This crypto-system uses modular arithmetic, historically it used modulo 26 since it was used for text information and the English alphabets contains 26 letters only. In this research, modulo 26 is replaced by modulo 256 as the intensity in the image can vary from 0 to 255 different levels.

For the RGB image, each layer is scanned vertically so that each 4 pixels can form a 4×1 vector. Those vectors are multiplied by the 4×4 self-invertible key matrix.

The Hill Cipher was implemented using the following steps:

1. Each layer of the RGB layers is converted to a vector of $n \times 1$ elements using vertical scan. Where n is the total number of pixels in the image obtained by multiplying the dimensions (row $\times$ column) as shown in Figure 3.5 below.
2. Padding the resultant vector $n \times 1$ with zeros if its length is not a multiple of 4. This is because the Hill Cipher is executed by matrix multiplication of the key, which is 4×4 , by the sector of the plaintext (image) which should be then 4×1 vector.

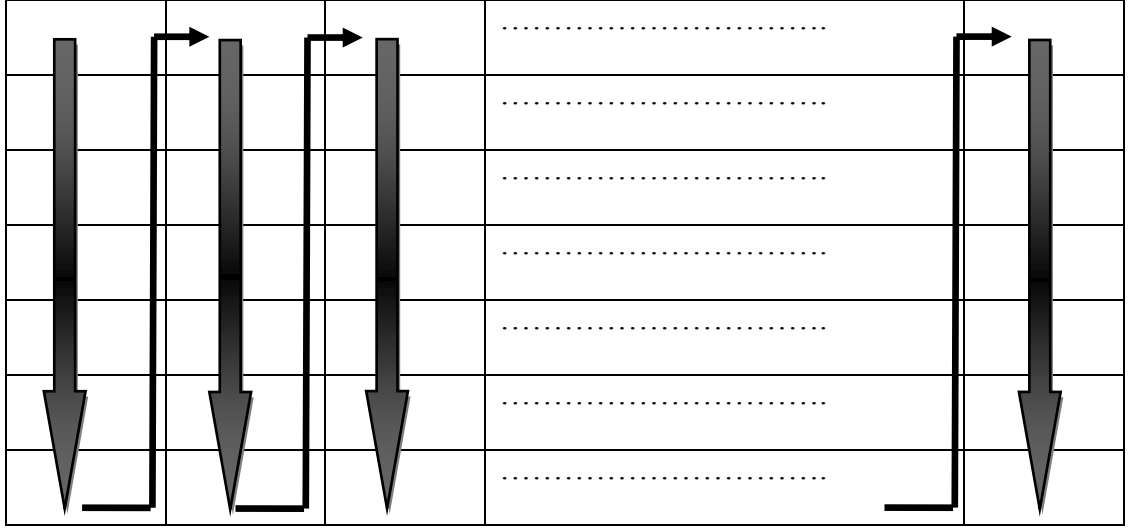


Figure 3.5: The vertical scanning used to change the image dimensions.

3. Take each four pixels from the resultant vector which represents plain image (P) and multiply it by the key matrix (K) using the following equation:

$$C = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \times \begin{bmatrix} P_{11} \\ P_{21} \\ P_{31} \\ P_{41} \end{bmatrix} \pmod{256} \quad (3.1)$$

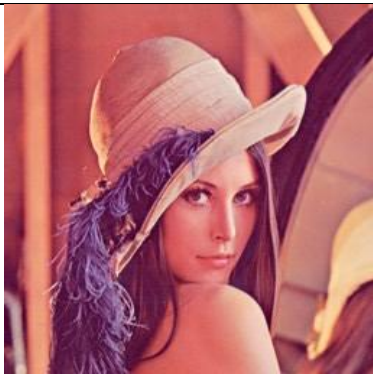
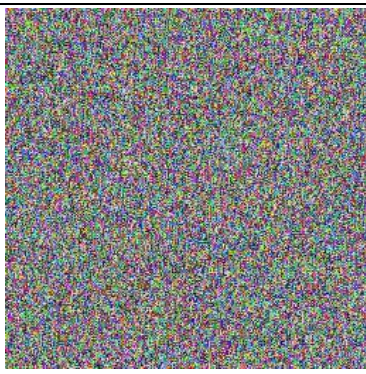
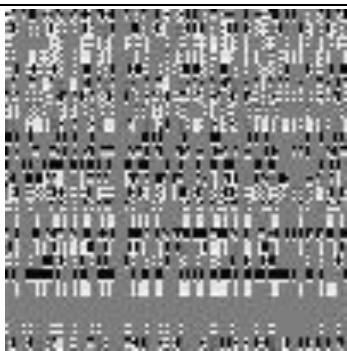
Where C is the 4×1 ciphered block of the image, P is a 4×1 block of the original image (Plain Image) and K is a 4×4 key matrix.

4. The ciphered block sector (C) is stored in the ciphered vector in the same location of the input sector (P) in the plain vector created in step 2 before.
5. The steps 3 and 4 are repeated for all equally divided sectors of the vector obtained from step 2.
6. The padded zeros, if exist, are removed after ciphering all the vector, which means that all the pixels in the image has been ciphered.
7. The ciphered vector is reshaped again to make the ciphered image dimensions as the same as the dimensions of the original image. So that the vertical scanning in step 2 is reversed.

Changing the dimensions of the image (or layer in RGB image) to make the column vectors (as in step 2 above) and returning the ciphered vector back to the original dimensions (as in step 7 above) was easy to implement in MATLAB using the "reshape" function. As a matter of fact, the "reshape" function in MATLAB change the matrix dimensions in a column wise fashion [15].

The following table (Table 3.3) shows the final ciphered image for two samples. The first sample contains an image of many details while the second sample has a firm white background with small black objects. The comparison here shows that the problem usually appeared in ciphering the images like the second sample in previous techniques has been solved.

Table 3.3: Final Results for the MHill Algorithm

Input image	Ciphered using the proposed MHill algorithm
	
01234 56789	

3.3 The proposed Decipher

It is clear that decipher is achieved by reversing the cipher procedure. So, to decrypt the ciphered image according to the proposed MHill algorithm, the receiver should apply the following steps:

1. The first step in decipher is to calculate the deciphering key, which is the inverse of the key matrix.
2. As the ciphering operation ends with Hill Cipher, the decipher starts with Hill decipher. It is the same operation of Hill Cipher (Equation 3.1) except that the key used in decipher is (K_D) rather than (K) and the plain image (P) will be replaced by the ciphered image (C).

So, if the received ciphered image is (C_R), the output of Hill decipher will be (P_R) according to the following formula:

$$P_R = K_D \times C_R \text{ mod } 256 \quad (3.3)$$

3. The output of this process (P_R) should then recover its original pixel location to reverse the random permutation process in the ciphering. So, the receiver should know the mapping table that was designed in the ciphering side. This operation was implemented simply in MATLAB by the following piece of code:

```
[r c]=size(Image);  
  
for i=1:r  
    for j=1:c  
        img(new_row(i),new_colomn(j))=Image(i,j);  
    end  
end
```

Where (Image) is the output from Hill de-cipher (P_R), (new_row) and (new_column) are vectors representing the tables of random permutation for rows and columns respectively. Figure 3.6 shows the flowchart of the proposed decipher process in the MHill algorithm.

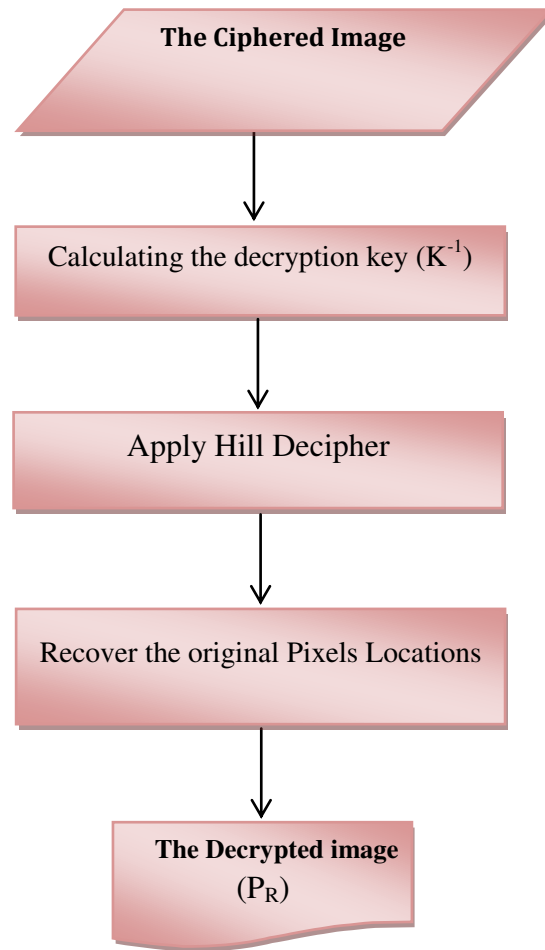


Figure 3.6: The proposed decipher process of the MHill algorithm

3.4 The MATLAB Simulation

The simulation program was designed to be a user friendly application. By running it, a file selection window will be popped up asking the user to choose the image that he needs to cipher. It gives the user the ability to choose between the "JPEG" and "BMP" file types as shown below.

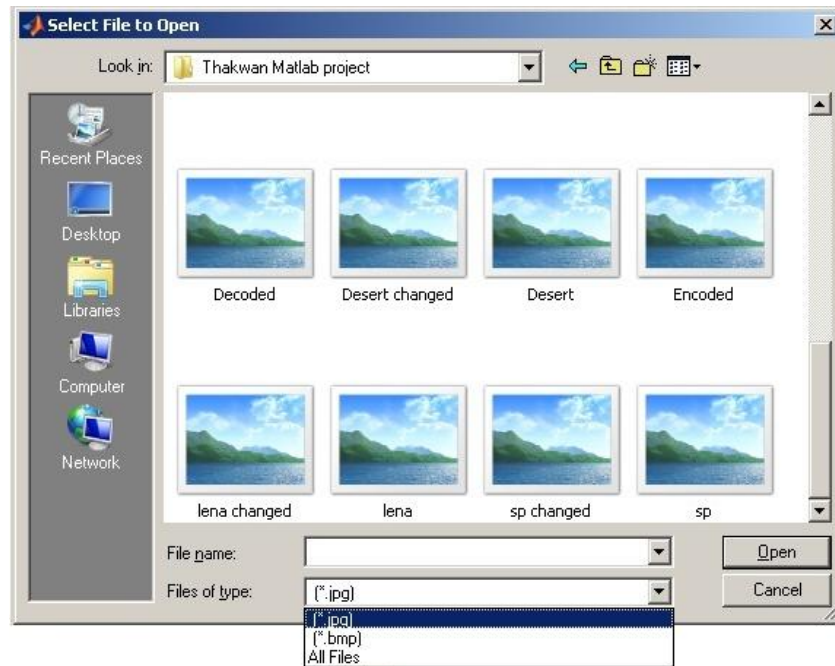


Figure 3.7: The image selection window used by the simulation

After choosing the image from Figure 3.7, the program will start ciphering and deciphering process sequentially. Then it will show the measurements of the ciphering elapsed time, correlation coefficient, similarity and irregular deviation factor in the command window of MATLAB. Besides, a figure window (shown in Figure 3.8) that shows the original (or plain), the ciphered and the deciphered images and their associated histograms will come into sight as shown below.

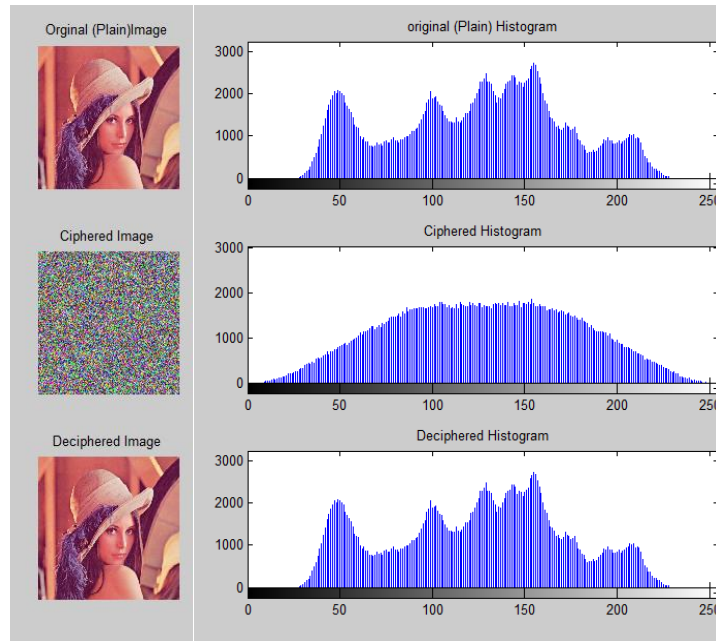


Figure 3.8: The result window of the simulation

Where the x-axis represents the gray-level value which has the range from zero to the number of gray levels. The y-axis represents the number of times of corresponding intensity value occurred in the image

Results and measurements will be discussed later in the next chapter. In this simulation only some of the measurements were taken while other measurements are calculated in separate programs which are also implemented using MATLAB software.

Chapter 4

RESULTS AND DISCUSSION

4.1 Introduction

Ciphering algorithm changes the pixels of the original image with compared with the ciphered image. These changes in pixels must be done in an irregular manner to disguise the transmitted ciphered image against the attacker. In addition, to achieve an acceptable ciphering, the ciphered image must be imperturbable of random patterns that do not expose any of the features of the original image. Besides, the ciphered image has to be uncorrelated with the original (plain) image [16-20].

The visual inspection can prove that the features of the plain image has been removed but it is not enough. Therefore, quantitative evaluation metrics should be used to assess the degree of encryption [22].

The results of using the proposed MHill technique were evaluated using different metrics including visual inspection. Those results were compared to the basic Hill image cipher on the same samples.

Following in this chapter, the quantitative measurements are discussed on many different samples.

4.2 Cipherring Metrics

The metrics that measure the quality of the cipherring techniques can be classified into two families [23]:

1. The first family measures the ability of the cipherring algorithm to produce an uncorrelated ciphered image. Three metrics are considered for this family which are the correlation coefficient (C_C), the histogram uniformity, and the irregular deviation (IDF).
2. The second family estimates the diffusion features of the encryption algorithm. Two metrics are considered for this family which are the Number of Pixels Change Rate (NPCR) and the Unified Average Change Intensity (UACI).

4.2.1 The Correlation Coefficient

The correlation measurement is typically used to measure how two signals are related or similar to each other. So, it could be said that it can measure the similarity of two different signals. Regarding this project, the correlation coefficient is used to measure the difference of two images.

This metrics used to assess the cipherring quality of any image cryptosystem. So that it calculates the correlation coefficient between pixels at the same locations in the plain and the ciphered images [24].

For two dimensional images, the correlation [15]:

$$C_C = \frac{|\sum_m \sum_n (P - \bar{P}) (C - \bar{C})|}{\sqrt{(\sum_m \sum_n (P - \bar{P})^2)} \sqrt{(\sum_m \sum_n (C - \bar{C})^2)}} \quad (4.1)$$

Where $\bar{P}$ is the two dimensional mean of the plain image (original) P , $\bar{C}$ is the two dimensional mean of the ciphered image C and both m and n are the row and column of the matrix respectively.

The cipher is better when C_C is close to zero since it indicates that the ciphered image is uncorrelated to the original image.

4.2.2 Similarity

It measures the similarity between the plain image P and the deciphered image P_R using the correlation coefficient between them. The higher the similarity is the better the algorithm.

4.2.3 Histogram Uniformity

The histogram represents the relation between the gray-level and the intensity of the image. Where the x-axis represents the gray-level value which has the range from zero to the number of gray levels. The y-axis represents the number of times of corresponding intensity value occurred in the image [25].

For ciphering image algorithms, two properties should be adopted [23]:

1. Both histograms of the original image and the ciphered image must be different.
2. Histogram of the ciphered image must be presented in a homogeneous distribution which leads to have an equal chances of presence for any intensity value in a random manner. This test was made by using visual inspection and the MATLAB built in function "*imhist*".

4.2.4 The Irregular Deviation Factor

The irregular deviation factor is a quality measurement of how much the deviation caused on the encrypted image by encryption algorithm is irregular.

Calculating this metric can be done by these steps [23]:

1. Find the absolute value (D) of the difference in pixels between the plain image (P) and the ciphered image (C).

$$D = |P - C| \quad (4.2)$$

2. Calculate the histogram H of this absolute difference matrix.

$$H = \text{histogram}(D) \quad (4.3)$$

3. Find the mean value M_H of this histogram

$$M_H = \frac{\sum_0^{255} H}{256} \quad (4.4)$$

4. Determine the absolute of the histogram deviations from this mean value as follows:

$$H_D(i) = |H(i) - M_H| \quad (4.5)$$

The irregular deviation factor *IDF* is calculated as follows:

$$IDF = \frac{\sum_0^{255} H_D(i)}{m \times n} \quad (4.6)$$

The lower the value of *IDF* is, the better the encryption quality.

4.2.5 Diffusion Metrics

When it comes to measure the diffusion characteristics in any cipher algorithm, the common metrics usually designed to measure the influence of one-pixel change on the whole image. Two common measures may be used; the Number of Pixels Change Rate (NPCR) and the Unified Average Change Intensity (UACI) [27].

Those metrics are designed as following [27]:

1. Compute two ciphered images CC_1 and CC_2 . CC_1 is the ciphered form of the plain image. While CC_2 comes from ciphering the same plain image but have only one pixel difference from the first plain image.

2. Obtain the gray-scale form of CC_1 and CC_2 . Let $C_1(i,j)$ and $C_2(i,j)$ be the gray scale values of both CC_1 and CC_2 respectively.
3. The UACI measures the average intensity of difference between the two images as follows [23]:

$$UACI = \frac{1}{m \times n} \left(\sum_{i,j} \frac{C_1(i,j) - C_2(i,j)}{255} \right) \times 100\% \quad (4.7)$$

4. To measure the NPCR, first define a matrix D , with the same size as the images C_1 and C_2 . Then, $D(i,j)$ is determined from $C_1(i,j)$ and $C_2(i,j)$, so that:

$$D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_2(i,j) \\ 1, & \text{Otherwise} \end{cases} \quad (4.8)$$

The NPCR is defined as:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100\% \quad (4.9)$$

When comparing two or more cipher algorithms, the algorithm of the higher values of UACI and NPCR is considered to be better than the others.

4.2.6 Execution Time

The execution time represents the time required for simulating the ciphering and deciphering of an image. It can be affected by the complexity of the applied algorithm and by specifications the computer system that runs the certain algorithm. It is obvious that the smaller the execution time is, the better the cipher algorithm. In this work, the computer that was used in the experiments was HP G62 Notebook PC of Intel® Core™ i3 CPU. The operating system was Windows 7 Ultimate 32-bit. The application used to simulate the MHill and the Hill Cipher was MATLAB version 7.8.0.347 (R2009a).

4.3 Testing the MHill Algorithm

The MHill and the conventional Hill algorithms were implemented using six different samples. The samples of interest were those of small objects with wide fixed backgrounds. However, other kinds of images had been considered as well. The simulation code had been ran ten times and the results of the performance metrics were taken into an Excel to find out the average of each result separately as shown in Table 4.1.

Table 4.1: MHill and Hill Quality Tests

Sample	Ciphering Algorithm	Correlation Metrics Family			Diffusion Metrics Family		Execution Time (Sec)
		Correlation Coefficient C_c	Similarity	IDF	NPCR %	UACI %	
lena.jpg	Hill Cipher	0.013	1	39764	0.89	0.003	0.26
	MHill Cipher	0.0018	1	28392	99.63	33.59	0.343
SP.jpg	Hill Cipher	0.164	1	113933	0.0026	0.007	0.58
	MHill Cipher	0.002	1	48728	99.61	33.54	0.74
manara.jpg	Hill Cipher	0.323	1	78438	0.0017	0.028	0.61
	MHill Cipher	0.0014	1	38289	99.61	33.46	0.95
Donald.bmp	Hill Cipher	0.4485	1	64782	0.0044	0.037	0.35
	MHill Cipher	0.0012	1	18871	99.78	33.16	0.47
image1.bmp	Hill Cipher	0.827	1	53284	0.04	0.077	0.048
	MHill Cipher	0.0023	1	15274	55.34	20.31	0.11
image2.bmp	Hill Cipher	0.81	1	58321	0.04	0.076	0.048
	MHill Cipher	0.0025	1	17262	66.93	23.47	0.11

Table 4.1 shows that there was a big difference between using the MHill and the Hill Cipher techniques. As some metrics show an advantage of the proposed MHill over the straight Hill Cipher, other shows the opposite. The final conclusion about the results in Table 4.1 above could be recapped in the following points:

1. The MHill Cipher produces a ciphered image that was completely different from the original (plain) image. This fact appear clearly by decreasing the correlation coefficients between the original and the ciphered images in the MHill compared with the Hill Cipher.
2. The IDF was improved by using the proposed MHill algorithm compared with using the conventional Hill Cipher.
3. The similarity between the original and the deciphered image in both MHill algorithm and the conventional Hill Cipher is equal to 1. So, the decipher process returned the original image as it was with little losses due to ciphering.
4. In the other hand, the diffusion measurements (NPCR and UACI) show that there is an advantage of using MHill rather than the Hill Cipher. However, there were some exceptions as the sample (image1.bmp and image2.bmp) as well as the results that were obtained by using the AdvHill Cipher because the color intensity of the original image is mainly distributed in only two regions at region zero and region of 255 while the whole region in between is not being implemented.
5. As a matter of fact, the expected NPCR value could be calculated using the following equation [30]:

$$expected_NPCR = (1 - 2^{-L}) \times 100\% \quad (4.10)$$

Where L is the number of bits used to represent each pixel in the image. In this case L is 8, so the theoretical expected NPCR is 99.6%.

6. It could be said in general that the histogram uniformity was achieved by the MHill process and it was better than the straight Hill image cipher.
7. Execution time in the above table is the cipher time only. Definitely, the MHill takes more time than the Hill algorithm as it is more complicated.

In general, the MHill Cipher technique is an improvement of the straight Hill as there was an improvement in both of the two families of cipher metrics. Firstly, the correlation coefficient C_C was decreased which means that the MHill produced an uncorrelated ciphered image. For the second family of metrics, the diffusion characteristics, both the NPCR and UACI measurements were improved.

The following figures show the histograms of all the samples for both the proposed MHill and the conventional Hill Cipher techniques. The tested samples in Figures 4.1 to 4.12 shows the advantages of using MHill in image cipher.

1. There was no significant improvement in ciphering the (lena.jpg) sample by using MHill. Nearly the same results obtained by using either the traditional Hill Cipher or even the two-stage or the proposed MHill Cipher for this sample. Although, there was a small difference in the ciphered histogram uniformity as shown in Figures 4.1 and 4.2. This is because this sample contains many details with graded intensity.

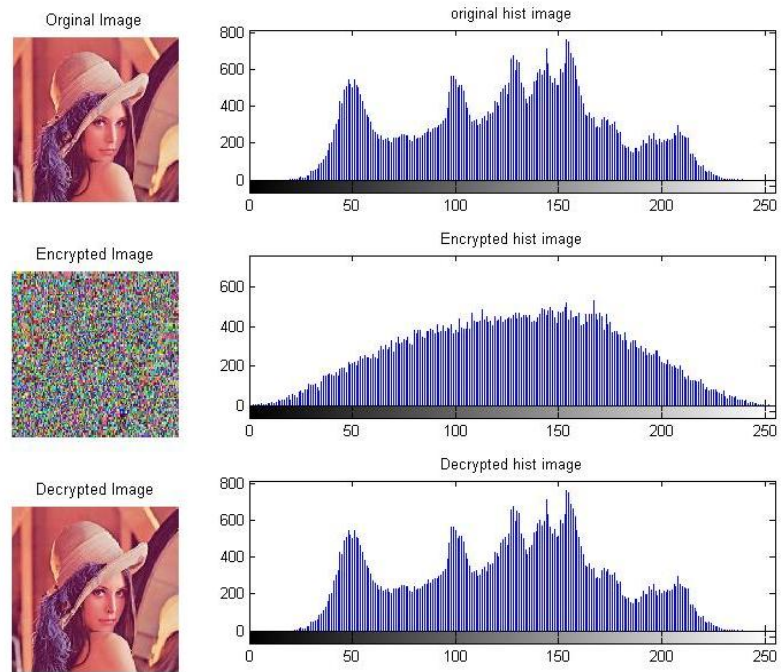


Figure 4.1: Hill Cipher Histogram for the sample lena.jpg

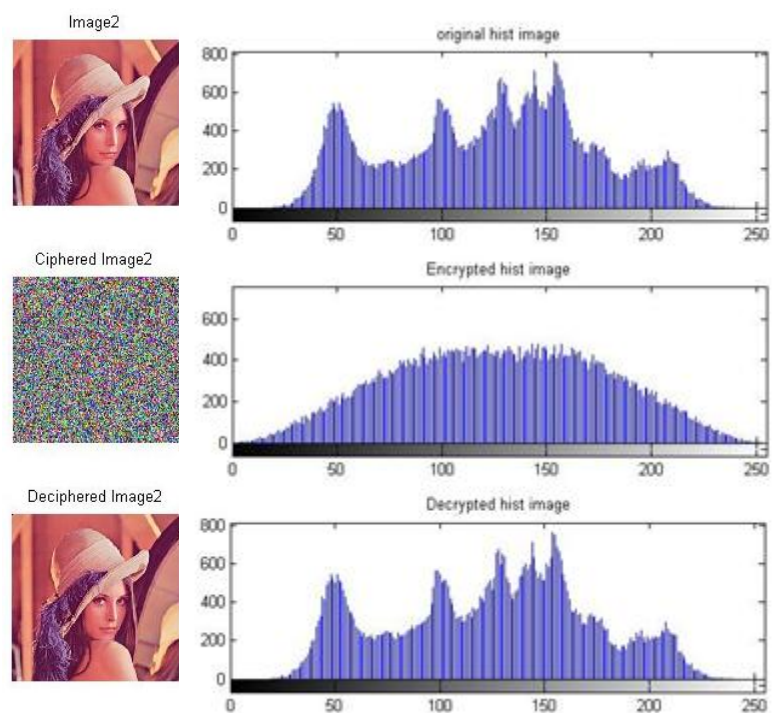


Figure 4.2: MHill Cipher Histogram for the sample lena.jpg

2. The second sample (SP.jpg) was ciphered using Hill in Figure 4.3 and then ciphered using MHill as shown in Figure 4.4 above. Here, ciphering using the MHill was more secure and efficient. This sample contains a black and relatively large background. This black background means a lot of zeros intensity levels. This is not comprehensible by the straight Hill Cipher and it was solved by the proposed algorithm.

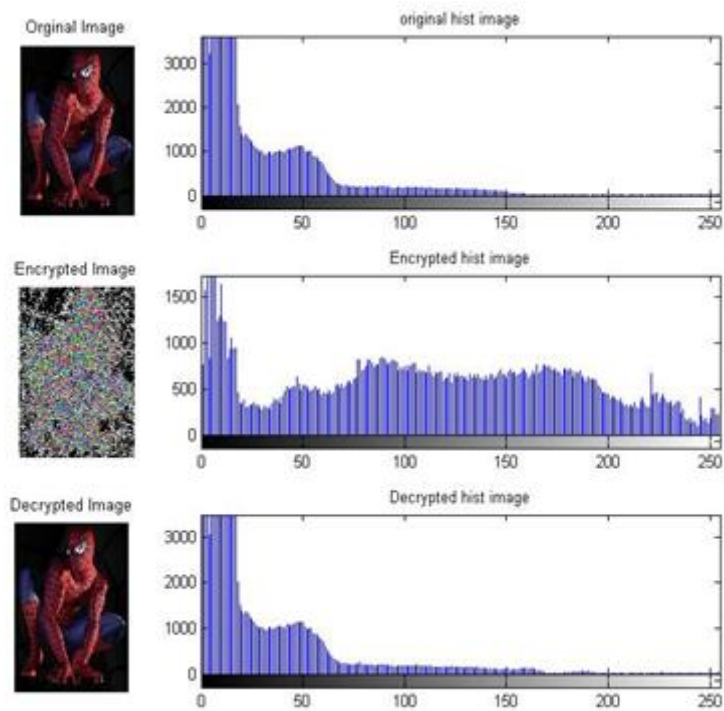


Figure 4.3: Hill Cipher Histogram for the sample SP.jpg

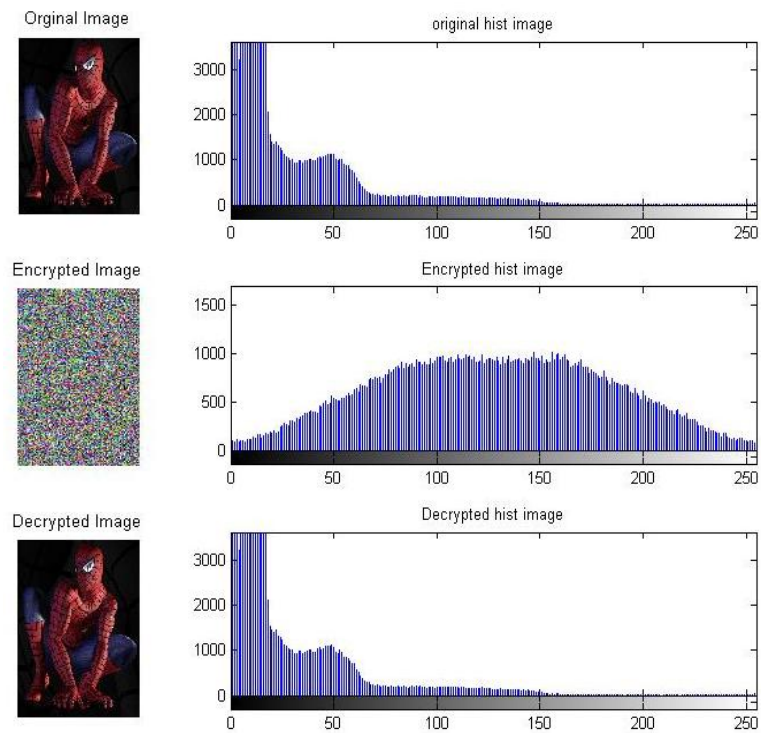


Figure 4.4: MHill Cipher Histogram for the sample SP.jpg

3. Figures 4.5 and 4.6 also show the improvement made by the proposed MHill strategy. This sample (manara.jpg) seems to be the same as the sample discussed in point 2 but the background is white. The MHill worked properly for this sample while the Hill did not cipher it well.

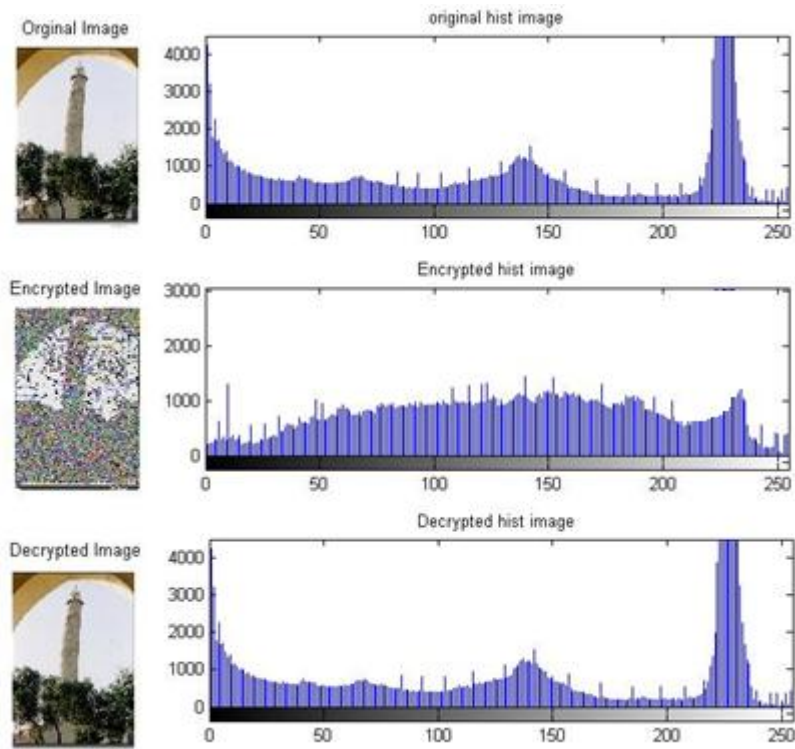


Figure 4.5: Hill Cipher Histogram for the sample manara.jpg

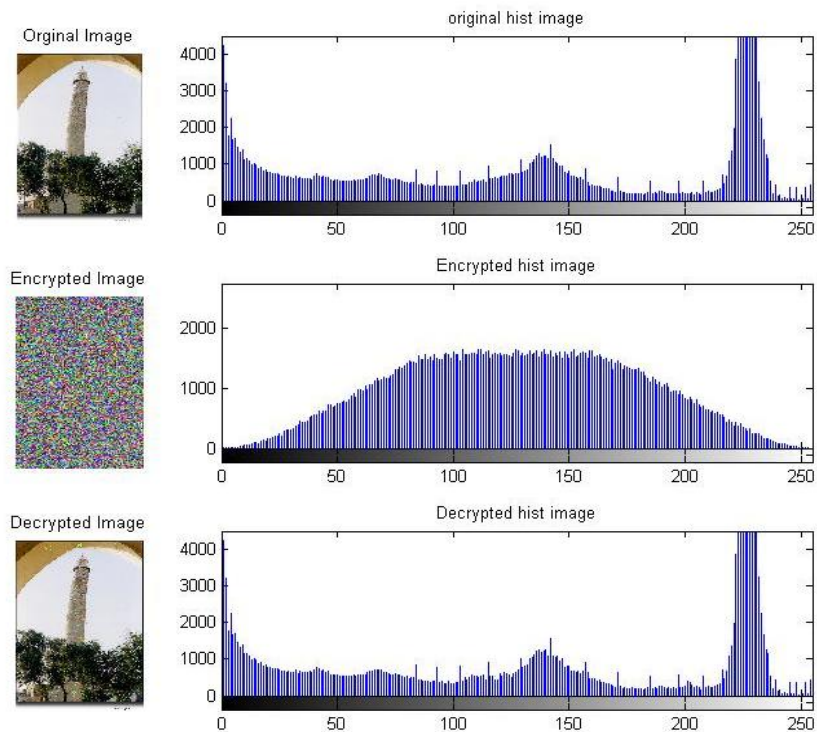


Figure 4.6: MHill Cipher Histogram for the sample manara.jpg

4. The sample showed in Figures 4.7 and 4.8 is an example of how the designed MHill algorithm beat the problem of image cipher. The wide background decreased the Hill Cipher security while using the proposed algorithm the security was not affected.

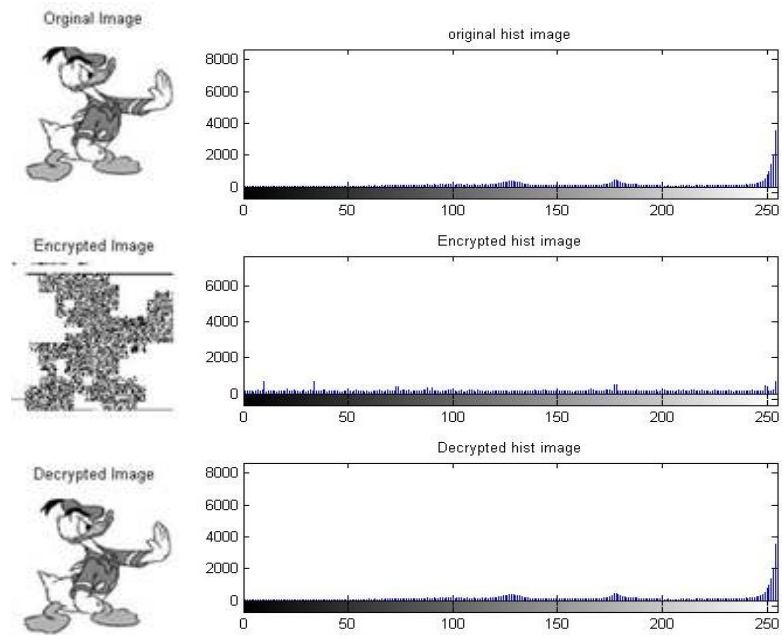


Figure 4.7: Hill Cipher Histogram for the sample donald.bmp

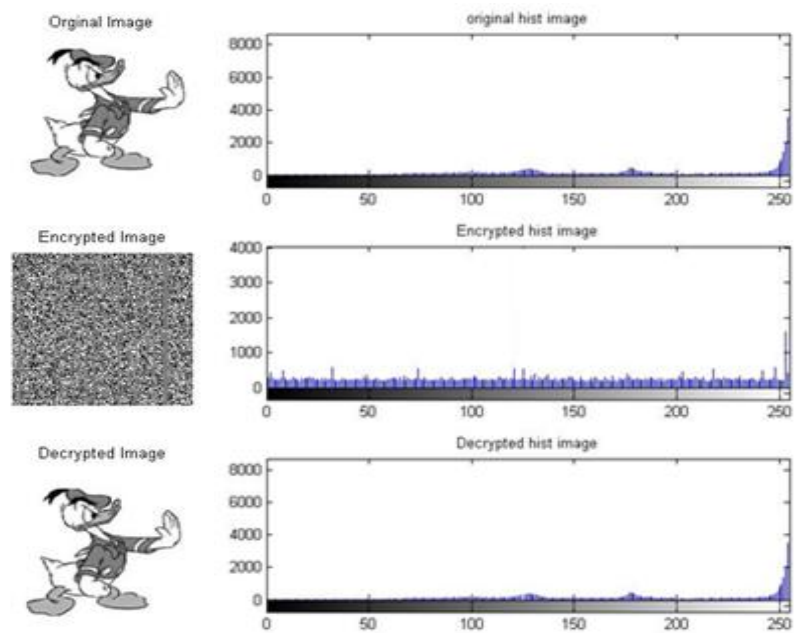


Figure 4.8: MHill Cipher Histogram for the sample donald.bmp

5. The more obvious example about the security of the MHill Cipher was the sample of an image that contained white backgrounds and black written text. By inspection Figures 4.9 and 4.11 where the traditional Hill Cipher was used, the written text in the image was still readable even after encryption. While the proposed MHill has proved its strength in the sense of hiding the text.

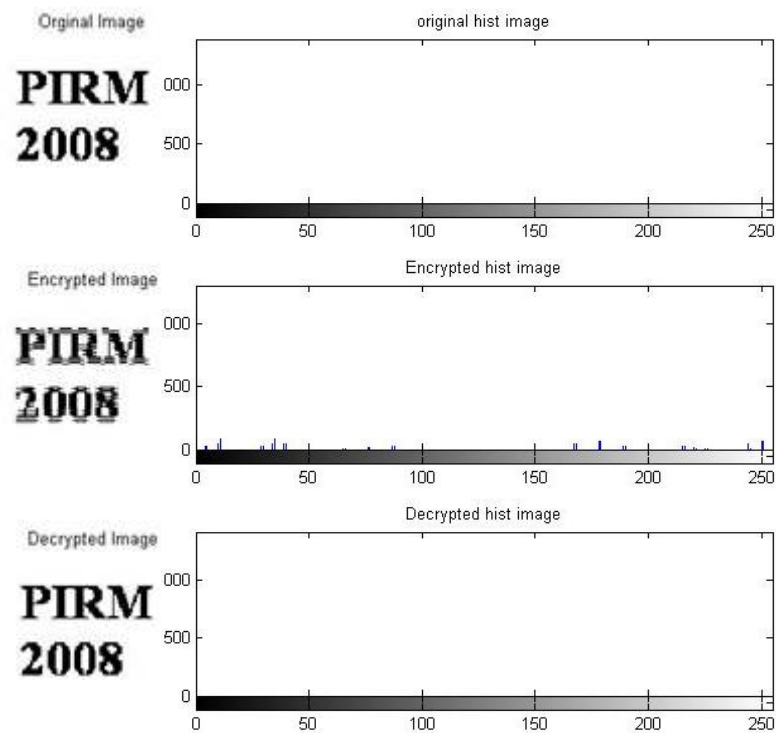


Figure 4.9: Hill Cipher Histogram for the sample image1.bmp

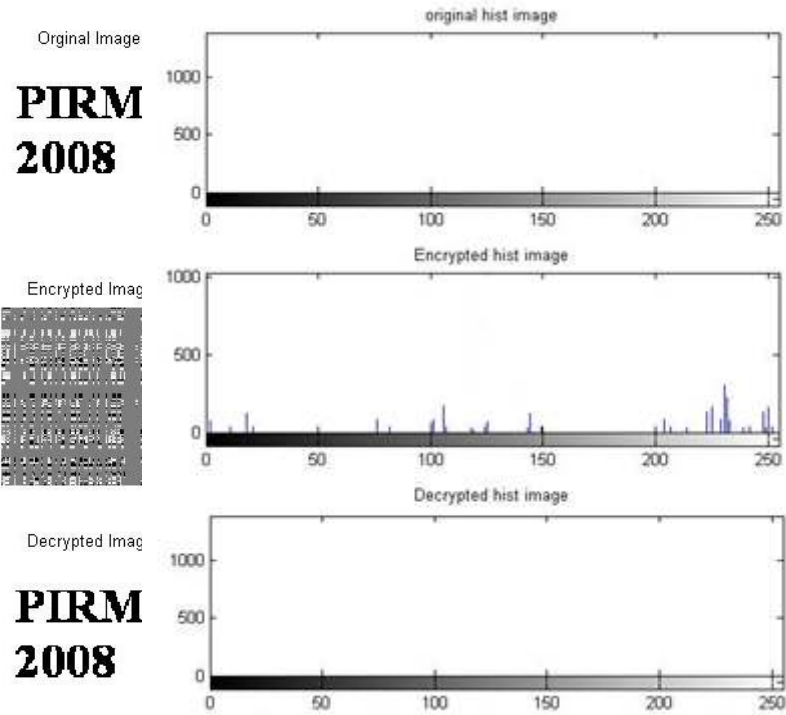


Figure 4.10: MHill Cipher Histogram for the sample image1.bmp

6. From the last results of the two samples (in Figures 4.11 to 4.12), it could be concluded that the proposed algorithm (MHill) hid the important information where the Hill Cipher failed. In fact, the proposed re-quantization and the random permutation steps in the MHill algorithm were the key steps of this improvement.

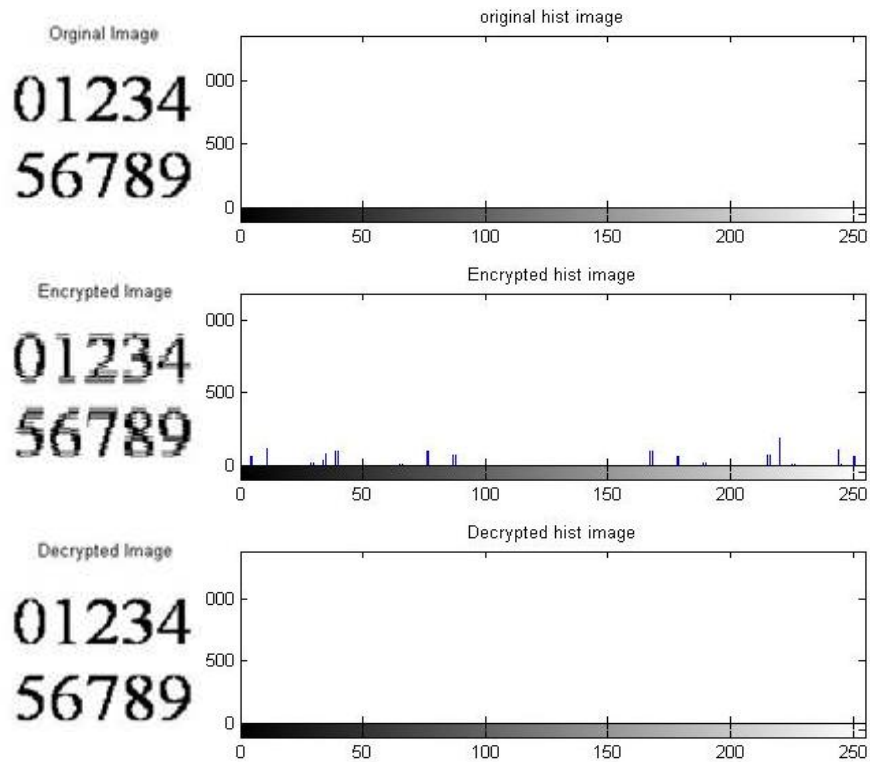


Figure 4.11: Hill Cipher Histogram for the sample image2.bmp

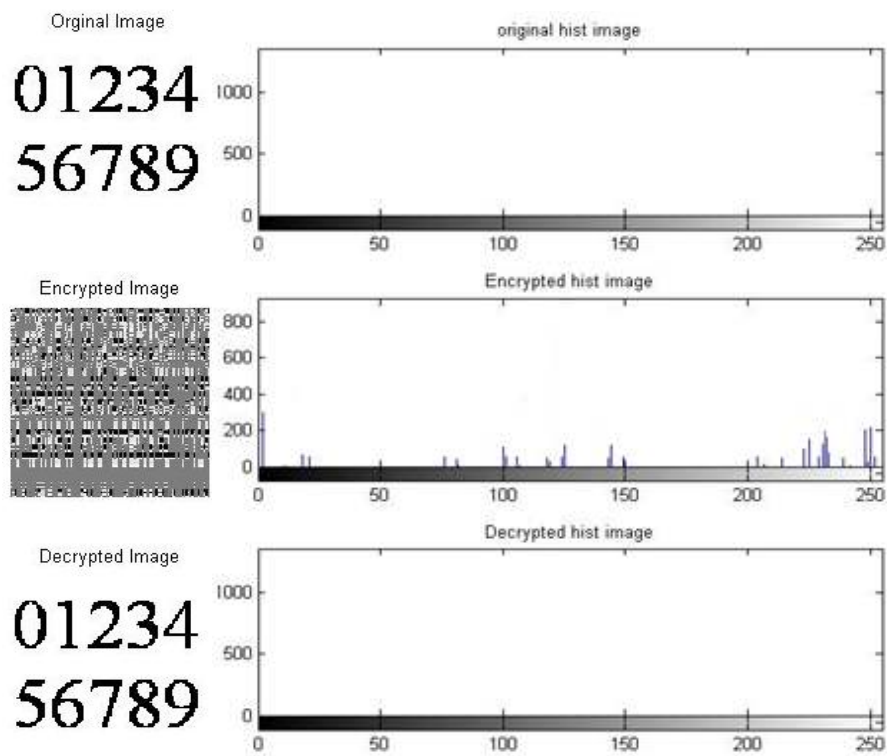


Figure 4.12: MHill Cipher Histogram for the sample image2.bmp

When it comes to compare the proposed MHill with previous works of other researchers, the first one seems to be better. For example, when Mutto *et al* cryptosystem [29] ciphered a (Donald.jpg) sample, the correlation coefficient (C_C) was (0.0499), while in our system it is (0.0044).

Another example of recent works is the "Advanced Partial Image Encryption using Two-Stage Hill Cipher Technique" paper that was published in 2012 by Panduranga H T and Naveen Kumar S K [9]. Although the application did not obligate them to cipher the whole image, they applied the Hill Cipher twice in their system and the maximum NPCR was 97.74%. In opposite to that algorithm is the proposed MHill in our work, where Hill Cipher is only applied once and the NPCR reaches 99.63% maximum. In addition; AdvHill that was proposed in "On the Privacy Protection of Biometric Traits: Palmprint, Face, and Signature" [10], works only when there were many details in the image and failed for small objects scenes. Table 4.2 shows comparisons between the proposed MHill and the Two-Stage Hill with the Standard Optimal Performance Expected Values.

Table 4.2: Results obtained by proposed MHill and Two-Stage Hill for Lena.jpg Image

 Lena.jpg	Similarity	NPCR %	UACI %
MHill	1	99.63	33.59
Two-Stage Hill	1	97.74	38.33
Standard Optimal	1	99.60	33.49

Chapter 5

CONCLUSION

The basic Hill Cipher uses a linear operations and modular mathematics and it is based on the fact that its power lies on its security. The ciphering key is a matrix, so the bruit-attacks will be weakened definitely. However, using this technique to cipher an image faced a problem of the inability for hiding the image details. Many researchers have either tried another types of ciphering techniques or modified it. In spite of the fact that there are other methodologies, the Hill technique still attract researchers because it is the famous symmetric key encryption technique due to its simplicity and security.

Roughly, all the Hill modifications are based on adding extra steps before applying it. Those steps usually designed to solve the problem of sustaining of the patterns and details of the image after applying the traditional Hill Cipher on it. The proposed MHill algorithm in this research also designed in this way, although it was intended to have less computational expenses.

The proposed MHill strategy that uses dynamic re-quantization with the help of permutation was compared with the conventional Hill. It gave promising results in sense of producing uncorrelated ciphered images and improving the diffusion characteristics. It can be noticed from Table 4.1 that conventional Hill failed in correlation coefficient between the original image and the ciphered image.

These algorithms have been implemented for image encryption. Quality of image encryption for all algorithms is studied using visual inspection and quality measurement factors. From the obtained results, it is observed that the proposed MHill is more effective in encryption quality in the case of images with large single color areas.

The proposed MHill resists the known plaintext-ciphertext attack because of the use of quantization and permutation experimental analysis also shows that the MHill resist the statistical attacks.

In future works, the security for the proposed MHill strategy could be upgraded if another random permutation process is added after the last step of the ciphering algorithm. Nevertheless, this extra operation will definitely make the cryptosystem more computationally expensive. The security could be enhanced by adding another extra permutation or increasing the key size.

REFERENCES

- [1] Imai H., Hanaoka G., Shikata J., Otsuka A., Nascimento A.C. "Cyptography with Information Theoretic Security". Information Theory Workshop 2002, Proceedings of the IEEE, pp. 20-25, Oct 2002.

- [2] Panigrahy S. K., Acharya B., Jena D., "Image Encryption Using Self-Invertible Key Matrix of Hill Cipher Algorithm", 1st International Conference on Advances in Computing, Chikhli, India, pp. 21-22, February 2008

- [3] Nordin A., Rahman A., Abidin A., Mohd K., Usop N., "Cryptography: A New Approach of Classical Hill Cipher". International Journal of Security and Its Applications, Volume 7, No. 2, March, 2013

- [4] Saeednia, S., "How to make the Hill Cipher secure". Cryptologia. Volume 24 No.4, pp. 353-360, 2000.

- [5] Stallings, W. "Cryptography and Network Security", 4th edition, Prentice Hall, ISBN-10: 0-13-187319-9, 2005.

- [6] Overbey J., Traves J., Wojdylo W., "On the key space of the Hill Cipher. Cryptologia". Volume 29, No. 1, pp. 59-72, 2005.

- [7] Sastry K., Ravi Sh, "Modified Hill Cipher for a Large Block of Plaintext with Interlacing and Iteration", Journal of Computer Science. Volume 4, No. 1, pp. 15-20, 2008.
- [8] Acharya B., Sharma M. D. , Tiwari S., Minz V. K. , "Privacy Protection of Biometric Traits using Modified Hill Cipher with Involutory Key and Robust Cryptosystem". Procedia Computer Science 2, pp. 242–247, ICEBT 2010.
- [9] Panduranga H. T., Naveen S. K., "Advanced Partial Image Encryption using Two-Stage Hill Cipher Technique", International Journal of Computer Applications. Volume 60, No.16, pp. 875 – 887, December 2012.
- [10] Saroj K., Debasish J., Sathya B., Sanjay J., "On the Privacy Protection of Biometric Traits: Palmprint, Face, and Signature". IC3 2009, CCIS 40, pp. 182–193, Springer-Verlag Berlin Heidelberg 2009.
- [11] Ismail, A. Amin, M. & Diab, H. How to Repair the Hill Cipher. Journal of Zhejiang University Science. Volume. 7, No. 12, pp. 2022-2030, (2006).
- [12] Chefranov A. G., "Secure Hill Cipher Modification SHC-M Proc". Of the First International Conference on Security of Information and Network (SIN2007) 7-10 May 2007, Gazimagusa (TRNC) North Cyprus, Elci, A., Ors, B., and Preneel, B (Eds) Trafford Publishing, Canada. pp. 34-37, 2007.

- [13] Kalman D., White J., "Polynomial equations and circulant matrices". Math. Monthly 108, pp. 821–840. 2010.

- [14] Krishna A.V.N., Madhuravani K., "A Modified Hill Cipher using Randomized Approach", I. J. Computer Network and Information Security, Volume 5, pp. 56-62. Published Online June 2012 in MECS (<http://www.mecs-press.org/>).

- [15] Image Processing Toolbox™ User's Guide, R2011b.

- [16] Bourbakis N., Alexopoulos C., "Picture Data Encryption Using SCAN Patterns, Pattern Recognition". Volume 25, No. 6, pp. 577-581. 1992.

- [17] Feng Y., Yu X., "A Novel Symmetric Image Encryption Approach Based on an Invertible Two-Dimensional Map, "The 35th Annual Conference of IEEE Industrial Electronics (IECON '09), pp. 1973-1978. Nov. 2009.

- [18] Kuo C. J., "Novel Image Encryption Technique and Its Application in Progressive Transmission", J. Electron. Imaging. Volume 2, No. 4, pp. 345-351, 1993.

- [19] Chang H. K., Liou J. L., "An Image Encryption Scheme Based on Quadtree Compression Scheme", In Proceedings of The International Computer Symposium, Taiwan, pp. 230-237, 1994.

- [20] Scharinger J., "Fast Encryption of Image Data Using Chaotic Kolmogrov Flow", J. Electronic Eng. Volume 7, No. 2, pp. 318-325, 1998.
- [21] Yen J. C., Guo J. I., "A New Image Encryption Algorithm and Its VLSI Architecture", In Proceedings of The IEEE Workshop Signal Processing Systems, pp. 430-437, 1999.
- [22] Yen J. C., Guo J. I., "A New Chaotic Key Based Design for Image Encryption and Decryption", Proceedings of the IEEE International Symposium Circuits and Systems. Vol. 4, pp. 49-52, 2000.
- [23] El-Ashry I. F. "Digital Image Encryption", A Thesis Submitted for The Degree of M. Sc. of Communications Engineering, Faculty of Electronic Engineering, Menofia University in Egypt, 2010.
- [24] Newton D E, "Encyclopedia of Cryptology", ABC-CLIO Inc, California, USA, 1997.
- [25] Cheng J., Zhang F., Yu K., Ma J., "The Dynamic and Double Encryption System Based on Two-Dimensional Image", International Conference on Computational Intelligence and Security (CIS '09), pp. 458- 462, 2009.
- [26] Yun-Peng Z., Wei L., Shui-Ping C., Zheng- JunZ., Xuan N., Wei-DiD., "Digital Image Encryption Algorithm Based on Chaos and Improved DES" , IEEE International Conference on Systems, Man and Cybernetics, pp. 474- 479, 2009.

- [27] Nien H. H., Changchien S. K., Wu S. Y., Huang C. K., "A New Pixel-Chaotic-Shuffle Method for Image Encryption". The 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 883-887, 17-20 Dec.2008.
- [28] Huang C. K., Hsu Y. H., Chen W. Y., Changchien S. K., Hung C. M., Liu C. H., TianY. R., "High Security Image Encryption By Two-Stage Process", The 7thInternational Conference on Information, Communications and Signal Processing (ICICS), pp. 1-5, 8-10 Dec. 2009.
- [29] Muttoo S.K., Aggarwal D., Ahuja B., "A Secure Image Encryption Algorithm Based on HillCipher System", Buletin Teknik Elektro dan Informatika (Bulletin of Electrical Engineering and Informatics). Volume 1, No.1, pp. 51-60, March 2012.
- [30] Ullagaddi V., "Development of Data Encryption Algorithms for Secure Communication Using Public Images" , A Thesis Submitted for The Degree of M. Sc. of Electrical Engineering, Graduate Faculty at the University of Toledo, USA, August 2012.

APPENDICES

Appendix A: The proposed MHill MATLAB Simulation Program

```
clear all,close all;

clc;
display('The Proposed MHill cipher for image')
%%zdisplay('Choose the image');
input('Press Enter to continue :::','s');
[FileName,PathName,FilterIndex] = uigetfile({'*.jpg','*.bmp'});
im=[PathName FileName];
I1=imread(im);
[r c k]=size(I1);

%% Generation of random key
n = 4;      %%% Size of the key
det_key=0;
GCD=0;
A=[12; 5; 0;60];
B=[0;0;0;0];
eq=0;
while(~(det_key && GCD==1 && eq))
    key = floor(256.*rand(n,n));
    det_key = mod(det(key),256);
    GCD = gcd(round(det_key),256);
    AA = mod(key*A,256);
    key_d=256-mod(det(key)*inv(key),256);    %% Extracting the
detection key
    B=round(mod(key_d*AA,256));
    if(A==B)
        eq=1;
    else
        eq=0;
    end
end

%% Generating the New locations for permutation
newr=randperm(r);
newc=randperm(c);

%% Mapping
tic
I1(find(I1==0))=1;
%% Random Permutation is applied for each layer (R ,G and B)

RI=I1(:, :, 1);
GI=I1(:, :, 2);
BI=I1(:, :, 3);

RI=rperm(RI,newr,newc);
GI=rperm(GI,newr,newc);
BI=rperm(BI,newr,newc);

I2(:, :, 1)=RI;
I2(:, :, 2)=GI;
I2(:, :, 3)=BI;
%% Encoding
enc_im=Hill_enc(I2,key);
```

```

toc
%% Decoding
dec_im1=Hill_dec(enc_im,key);

%% Random De-Permutation is applied for each layer (R ,G and B)
RI=dec_im1(:,:,1);
GI=dec_im1(:,:,2);
BI=dec_im1(:,:,3);

RI=de_rperm(RI,newr,newc);
GI=de_rperm(GI,newr,newc);
BI=de_rperm(BI,newr,newc);

dec_im(:,:,1)=RI;
dec_im(:,:,2)=GI;
dec_im(:,:,3)=BI;

%% Shwoing results
subplot(3,2,1),imshow(I1);title('Original (Plain) Image')
subplot(3,2,2),imhist(rgb2gray(I1));title('original (Plain) Histogram')
subplot(3,2,3),imshow(enc_im);title('Ciphared Image')
subplot(3,2,3),imshow(enc_im);title('Ciphared Image')

subplot(3,2,4),imhist(rgb2gray(enc_im));title('Ciphared Histogram')
subplot(3,2,5),imshow(dec_im);title('Deciphared Image')
subplot(3,2,6),imhist(rgb2gray(dec_im));title('Deciphared Histogram')
imwrite(enc_im,'Encoded.jpg','jpg');

%% Measurements
fprintf('\n\n\n Some measurements \n\n\n')
im2=double(rgb2gray(I1));
enc_im2=double(rgb2gray(enc_im));
dec_im2=double(rgb2gray(dec_im));
Correlation_Coeefeicient=abs(corr2(im2,enc_im2))
similarity_Original_dec=corr2(im2,dec_im2)
[m n]=size(I1);
DI=abs(I1-enc_im);
DI=rgb2gray(DI);
h=imhist(DI)*256;
DC=sum(h)/256;
AC=abs(h-DC)*256;
Irregular_Deviation_Factor=sum(AC)/(m*n)
%%

fprintf('\n\n\n Wait for NPCR and UACI measurements \n\n\n')
%% Change one pixel
I=I1;
[m n l]=size(I);
I1(round(m/2),round(n/2),1)=255-(I(round(m/2),round(n/2),1));
I1(round(m/2),round(n/2),2)=255-(I(round(m/2),round(n/2),2));
I1(round(m/2),round(n/2),3)=255-(I(round(m/2),round(n/2),3));

%% Generating the New locations for permutation
newr=randperm(r);
newc=randperm(c);

%% Mapping
tic

```

```

I1(find(I1==0))==1;
%% Random Permutation is applied for each layer (R ,G and B)
RI=I1(:, :, 1);
GI=I1(:, :, 2);
BI=I1(:, :, 3);

RI=rperm(RI,newr,newc);
GI=rperm(GI,newr,newc);
BI=rperm(BI,newr,newc);

I2(:, :, 1)=RI;
I2(:, :, 2)=GI;
I2(:, :, 3)=BI;
%% Encoding
enc_im_ch=Hill_enc(I2, key);

%% NPCR
% I2=rgb2gray(enc_im);
% I3=rgb2gray(enc_im_ch);

[m n,c]=size(enc_im_ch);

for k=1:c
for i=1:m
    for j=1:n
        if(enc_im(i,j,k)==enc_im_ch(i,j,k)) D(i,j,k)=0;
        else
            D(i,j,k)=1;
        end
    end
end
end
NPCR=(sum(sum(sum(D))/(m*n*c)))*100

%% UACI
UACI=(2*(sum(sum(sum(abs(enc_im-enc_im_ch))/(255))/(m*n*c)))*100

```

Appendix B: The Proposed MHill MATLAB Simulation Program for Highly detailed images only (No De-quantization operation)

```
clear all,close all;
clc;
display('The Proposed MHill Cipher for image')
display('Choose the image')
input(' Press Enter to continue :::','s');
[FileName,PathName,FilterIndex] = uigetfile({'*.jpg'; '*.bmp'});
im=[PathName FileName];
I1=imread(im);
[r c k]=size(I1);
%% Generation of random key
det_key=0;
GCD=0;
while ~(det_key && GCD==1)
    key = floor(256.*random('Normal',0,255,4,4));
    det_key = mod(det(key),256);
    GCD = gcd(det_key,256);
end
%% Generating the New locations for permutation
newr=randperm(r);
newc=randperm(c);

%% Mapping
tic
% No Mapping
%% Random Permutation is applied for each layer (R ,G and B)

RI=I1(:, :, 1);
GI=I1(:, :, 2);
BI=I1(:, :, 3);

RI=rperm(RI,newr,newc);
GI=rperm(GI,newr,newc);
BI=rperm(BI,newr,newc);

I2(:, :, 1)=RI;
I2(:, :, 2)=GI;
I2(:, :, 3)=BI;
%% Encoding
enc_im=Hill_enc(I2,key);

toc
%% Decoding
dec_im1=Hill_dec(enc_im,key);

%% Random De-Permutation is applied for each layer (R ,G and B)
RI=dec_im1(:, :, 1);
GI=dec_im1(:, :, 2);
BI=dec_im1(:, :, 3);

RI=de_rperm(RI,newr,newc);
GI=de_rperm(GI,newr,newc);
BI=de_rperm(BI,newr,newc);

dec_im(:, :, 1)=256-RI;
```

```

dec_im(:,:,2)=256-GI;
dec_im(:,:,3)=256-BI;

%% De-Mapping

% No
%% Shwoing results
subplot(3,2,1),imshow(I1);title('Original (Plain) Image')
subplot(3,2,2),imhist(rgb2gray(I1));title('original (Plain) Histogram')
subplot(3,2,3),imshow(enc_im);title('CIPHERED Image')
subplot(3,2,4),imhist(rgb2gray(enc_im));title('CIPHERED Histogram')
subplot(3,2,5),imshow(dec_im);title('DecIPHERED Image')
subplot(3,2,6),imhist(rgb2gray(dec_im));title('DecIPHERED Histogram')
imwrite(enc_im,'Encoded.jpg','jpg');

%% Measurements
fprintf('\n\nSome measurements \n\n\n')
im2=double(rgb2gray(I1));
enc_im2=double(rgb2gray(enc_im));
dec_im2=double(rgb2gray(dec_im));
Correlation_Coefficient=corr2(im2,enc_im2)
similarity_Original_dec=corr2(im2,dec_im2)

[m n]=size(im2);
DI=abs(im2-enc_im2);
h=imhist(DI);
DC=sum(h)/256;
AC=abs(h-DC);
Irregular_Deviation_Factor=sum(AC)/(m*n)

```

Appendix C: The Hill MATLAB Simulation Program

```
clear all,close all;
clc;
display('Hill Cipher for image')
display('Choose the image')
input(' Press Enter to continue :::','s');
[FileName,PathName,FilterIndex] = uigetfile({'*.jpg'; '*.bmp'});
im=[PathName FileName];
I=imread(im);

%% Generation of random key
key=[5 35 252 221 ; 245 179 11 78; 6 35 251 221 ; 245 180 11 77];

%% Encoding
tic
enc_im=Hill_enc(I,key);
toc
%% Decoding
dec_im=Hill_dec(enc_im,key);

%% Shwoing results
subplot(3,2,1),imshow(I);title('Original Image')
subplot(3,2,2),imhist(rgb2gray(I));title('original hist image')
subplot(3,2,3),imshow(enc_im);title('Encrypted Image')
subplot(3,2,4),imhist(rgb2gray(enc_im));title('Encrypted hist
image')
subplot(3,2,5),imshow(dec_im);title('Decrypted Image')
subplot(3,2,6),imhist(rgb2gray(dec_im));title('Decrypted hist
image')
imwrite(enc_im,'Encoded.jpg','jpg');

%% Measurements
fprintf('\n\n\n Some measurements \n\n\n')
im2=double(rgb2gray(I));
enc_im2=double(rgb2gray(enc_im));
dec_im2=double(rgb2gray(dec_im));
Correlation_Coeefeicient=corr2(im2,enc_im2)
similarity_Original_dec=corr2(im2,dec_im2)
[m n]=size(im2);
DI=abs(im2-enc_im2);
h=imhist(DI);
DC=sum(h)/256;
AC=abs(h-DC);
Irregular_Deviation_Factor=sum(AC)./(m*n)
```

Appendix D: The Designed Functions that are used in programs in appendices A, B and C

Hill Encryption Function

```
function [enc_im]=Hill_enc(I,key)

[m n s]=size(I);
p=m*n;

x_R=reshape(I(:, :, 1),p,1);
x_G=reshape(I(:, :, 2),p,1);
x_B=reshape(I(:, :, 3),p,1);

added=0;
while(mod(p+added,4))
    x_R(end+1)=0;
    added=added+1;
end
l=length(x_R);
x_G(end+1:end+added-1)=0;
x_B(end+1:end+added-1)=0;
enc_x_R=zeros(p,1);
enc_x_G=zeros(p,1);
enc_x_B=zeros(p,1);

for i=1:4:l
    enc_x_R(i:i+3)=mod(key*double(x_R(i:i+3)),256);
end
for i=1:4:l
    enc_x_G(i:i+3)=mod(key*double(x_G(i:i+3)),256);
end
for i=1:4:l
    enc_x_B(i:i+3)=mod(key*double(x_B(i:i+3)),256);
end

enc_xx_R=enc_x_R(1:end-added);
enc_xx_G=enc_x_G(1:end-added);
enc_xx_B=enc_x_B(1:end-added);

enc_im_R=uint8(reshape(enc_xx_R,m,n));
enc_im_G=uint8(reshape(enc_xx_G,m,n));
enc_im_B=uint8(reshape(enc_xx_B,m,n));

enc_im(:, :, 1)= enc_im_R;
enc_im(:, :, 2)= enc_im_G;
enc_im(:, :, 3)= enc_im_B;
end
```

Hill decryption Function

```
function [dec_im]=Hill_dec(enc_im,key)
%% Decryption
key_d=256-mod(det(key)*inv(key),256);    %% Extracting the detection
key
%key_d=mod(det(key)*inv(key),256);    %% Extracting the detection
key
[m n s]=size(enc_im);
p=m*n;
x_decR=reshape(enc_im(:,:,1),p,1);
x_decG=reshape(enc_im(:,:,2),p,1);
x_decB=reshape(enc_im(:,:,3),p,1);

added=0;
while(mod(p+added,4))
    x_decR(end+1)=0;
    added=added+1;
end
l=length(x_decR);

x_decG(end+1:end+added-1)=0;
x_decB(end+1:end+added-1)=0;

dec_xR=zeros(p,1);
dec_xG=zeros(p,1);
dec_xB=zeros(p,1);

for i=1:4:l
    dec_xR(i:i+3)=mod(key_d*double(x_decR(i:i+3)),256);
end
for i=1:4:l
    dec_xG(i:i+3)=mod(key_d*double(x_decG(i:i+3)),256);
end
for i=1:4:l
    dec_xB(i:i+3)=mod(key_d*double(x_decB(i:i+3)),256);
end

dec_xxR=dec_xR(1:end-added);
dec_xxG=dec_xG(1:end-added);
dec_xxB=dec_xB(1:end-added);

dec_imR=uint8(reshape(dec_xxR,m,n));
dec_imG=uint8(reshape(dec_xxG,m,n));
dec_imB=uint8(reshape(dec_xxB,m,n));

dec_im(:,:,1)=dec_imR;
dec_im(:,:,2)=dec_imG;
dec_im(:,:,3)=dec_imB;

end
```

Random Permutation Function

```
function [img]=rperm(Image,new_row,new_column)

%% Random Permutation
[r c]=size(Image);

for i=1:r
for j=1:c
    img(i,j)=Image(new_row(i),new_column(j));
end
end
end
```

De-permutation Function

```
function [img]=de_rperm(Image,new_row,new_column)
%% De-permutation
[r c]=size(Image);
for i=1:r
for j=1:c
    img(new_row(i),new_column(j))=Image(i,j);
end
end
end
```

Appendix E: The Designed Programs to Measure NPCR and UACI Metrics

1- NPCR Program

```
function []=NPCR_Test
clear all,close all;
display('Choose the original encoded image')
input('Press Enter to continue :::','s');
[FileName,PathName,FilterIndex] = uigetfile({'*.jpg'; '*.bmp'});
im=[PathName FileName];
I=imread(im);
I1=rgb2gray(I);

display('Choose the encoded image with one pixel is changed from the
original')
input('Press Enter to continue :::','s');

[FileName,PathName,FilterIndex] = uigetfile({'*.jpg'; '*.bmp'});
im=[PathName FileName];
I=imread(im);
I2=rgb2gray(I);
[m n]=size(I2);
for i=1:m
for j=1:n
if (I1(i,j)==I2(i,j)) D(i,j)=0;
else
D(i,j)=1;
end
end
end
NPCR=(sum(sum(D))/(m*n))*100
end
```

2- UACI Program

```
clear all,close all;
display('Choose the original encoded image')
input('Press Enter to continue :::','s');
[FileName,PathName,FilterIndex] = uigetfile({'*.jpg'; '*.bmp'});
im=[PathName FileName];
I=imread(im);
I1=rgb2gray(I);

display('Choose the encoded image with one pixel is changed from the
original')
input('Press Enter to continue :::','s');
[FileName,PathName,FilterIndex] = uigetfile({'*.jpg'; '*.bmp'});
im=[PathName FileName];
I=imread(im);
I2=rgb2gray(I);

[m n]=size(I2);

UACI=((sum(sum(I1-I2))/255)/(m*n))*100
```