Development of Matrix Cipher Modifications and Key Exchange Protocol

Ahmed Yehya Mahmoud

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

Doctor of Philosophy
in
Computer Engineering

Eastern Mediterranean University
January 2012
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 Doctor of Philosophy in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
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 Doctor of Philosophy in Computer Engineering.

Assoc. Prof. Dr. Alexander Chefranov
Supervisor

Examining Committee

1. Prof. Dr. Evgueni Doukhnitch
2. Assoc. Prof. Dr. Alexander Chefranov
3. Assoc. Prof. Dr. Zeki Bayram
4. Asst. Prof. Dr. Gürcü Öz
5. Asst. Prof. Dr. Önsen Toygar

ABSTRACT

In modern cryptographic methods, keys are the basis for secure communication channels and the establishment of secret keys is a challenging problem for the large-scale deployment of symmetric cryptography to control encryption and decryption. Key establishment protocols provide exchanging secret information between two or more parties, typically for subsequent use as symmetric keys for a variety of information security services including encryption, message authentication, and entity authentication. They may be broadly subdivided into key transport and key exchange. Notably, key exchange is one of the difficulties when using symmetric algorithms, the key exchange particularly useful from a security viewpoint, for each of the key-sharing parties can have its own control and a high confidence on the quality of the key output. Beside encryption, key exchange is one of the most basic problems in cryptography; it becomes another challenge in cryptography.

This thesis is concerned with the modifications of the Hill cipher (HC), extension of Diffie Hellman and ElGamal key exchange protocols. The HC is one of the most popular symmetric key algorithms; it is resistant to brute-force and statistical attacks, but it can be broken with a known plaintext-ciphertext attack (KPCA). To overcome this vulnerability, several researchers tried to propose modifications of the Hill cipher and make it secure. However in the literature, most of these modifications are found to be either insecure or ineffective for image encryption.

The Diffie-Hellman Key Exchange (DH) is known as one of the public key algorithms, its aim is to distribute the keys over insecure channels. It is based on the complexity of discrete logarithm problem (DLP) solving over a finite field$GF(p)$, where p is prime which considered as an advantage from the security viewpoint due to the challenging and difficulties for solving the discrete logarithm. But DH has drawbacks including the fact that there are heavy and expensive exponential operations in both sides (sender and receiver) which affect its efficiency; it can be used for exchanging secret keys. To overcome this drawback, DH protocol matrix oriented modifications based on DLP are proposed by several researchers. Moreover, in the literatures, most of the modifications still rely on the DLP.

The ElGamal Public Key Cryptosystem and Signature (EPKCS) also rely on the computational complexity of finding discrete logarithms based on some publicly known primitive root (base element), $\alpha \in GF(p)$, where p is a large prime. Similar to DH protocol, the EPKCS has a drawback; it has a slow speed especially for signing in addition to the ciphertext is twice as long as the plaintext.

In this thesis, we proposed two modifications of the Hill cipher, HCM-EE and HCM-PRE. A matrix-based Diffie-Hellman-like key exchange protocol is also proposed. ElGamal public key cryptosystem and signature scheme is extended to the group *GU(m, p, n)* of numbers co-prime to mp^n.

Keywords: matrix cipher, dynamic key, image encryption, Diffie-Hellman key-exchange protocol, secure key-exchange protocol, ElGamal public key cryptosystem.

ÖZ

Modern şifreleme yöntemlerinde güvenli iletişim kanallarının temeli anahtarlardır. Simetrik şifreleme sistemlerinin geniş çaplı dağıtımında, gizli anahtarların iletişimi güçlük çıkartmaktadır. Anahtar tahsis protokolleri, gizli bilginin iki yada daha fazla taraf arasında iletişimini sağlamaktadır. Özellikle bu gizli bilgiler simetrik şifreleme anahtarları olarak; şifreleme, mesaj ve kimlik doğrulama gibi güvenli veri servislerinde kullanılmaktadır. Anahtar tesis etme sistemleri, anahtar taşıma ve değişimi olarak kabaca ikiye ayrılırlar. Simetrik algoritmalar kullanırken, özellikle anahtar değişimi zor olmaktadır. Güvenlik açısından bakıldığında, anahtar değişimi oldukça faydalıdır; anahtar paylaşan taraflar ortaya çıkan anahtarda pay sahibi olarak, güvenirliğinden emin olabilirler. Şifrelemeden sonra anahtar paylaşımı kriptografideki en temel problemdir.

Bu tez Hill şifrelemesi (HC) üzerine yapılan, Diffie Hellman ve Elgamal anahtar değişim protokolleriyle ilgilenmektedir. Hill şifrelemesi, en çok tercih edilen simetrik şifrelemedir. İstatistiksel ve zorlama saldırılarına karşı dayanıklı olmasına rağmen, bilinen salt metin - şifrelenmiş metin saldırısıyla kırılabilir. Bu açığı gidermek amaçlı Hill şifrelemesi üzerine bir çok değişiklik önerildi. Ancak, literatürdeki bu yöntemler resim şifrelemek için yetersiz kalmaktadır.

Diffie-Hellman Anahtar Değişimi (DH), güvenli olmayan yollardan anahtar dağıtımını sağlayan, açık anahtarlı bir algoritmadır. Bu algoritma, ayrık logaritmanın $GF(p)$ sınırlı alanı üzerinde çözümünün zorluğuna dayanmaktadır. Ancak, DH algoritmasının bazı sorunları mevcuttur. Bunlardan birisi, iki tarafta da yapılması gereken ve zaman alan üs alma işlemleridir. Bu sorunun üstesinden gelmek için, matrislere yönelik ayrık algoritma tabanlı değişiklikler öne sürülmüştür. Şu an literatürde ayrık logaritma kullanan bir çok çalışma mevcuttur.

Elgamal Açık Anahtarlı Şifre ve İmza Sistemi (EPKCS) de ayrık logaritmanın zorluğuna dayanmaktadır. Bu sistemlerde temel kök tabanb elemanı, herkes tarafından bilinmektedir; ve $\alpha \in GF(p)$ şeklinde belirtilmektedir. Burada p büyük bir asal sayıdır. DH algoritmasına benzer şekilde EPKCS'ın da sorunları mevcuttur. En önemli sorunu yavaş olmasıdır; özellikle şifrelenmiş metnin salt metnin iki katı olması bu sorunu artırmaktadır.

Bu tezde, Hill şifrelemesi üzerine HCM-EE ve HCM-PRE isimli iki değişiklik önerilmiştir. Matris tabanlı Diffie-Hellman-benzeri anahtar değişim protokolü de önerilmiştir. ElGamal Açık Anahtarlı Şifre ve imza Sistemi mp^{n}'e asal olan $GU(m,p,n)$ sayılarına genişletilmiştir.

Anahtar kelimeler: matris şifreleme, dinamik anahtar, resim şifreleme, Diffie-Hellman anahtar değişim protokolü, güvenli anahtar değişim protokolü, ElGamal açık anahtar şifre sistemi

To the soul of my mother

ACKNOWLEDGMENTS

I would like to thank Assoc. Prof. Dr. Alexander Chefranov, my supervisor, for many valuable suggestions and constant support during this research, without his invaluable supervision, all my efforts could have been short-sighted.

I had the pleasure of meeting Prof. Dr. Hasan Kömürcügil, previous Chairman of the Department of Computer Engineering, Eastern Mediterranean University. I am also thankful to Assoc. Prof. Muhammad Salamah, Chairman of Computer Engineering Department for his continuous support; I would like to thank all the staff of computer engineering department especially Prof. Omar Ramadan for his continuous encouragement.

I should not forget to express my thanks to a number of friends had always been around to support me morally. Beside my colleagues in the department, I would like to thank them as well.

I should also mention that my Ph.D. study in Turkish Republic of Northern Cyprus was supported in part by Al-Azhar University Gaza-Palestine.

I am indebted to my wife and my kids Yehya, Mohammed, Hala, Ibrahim, and Elina for their patience and *love*; I hope they will forgive me for their time. I would like to dedicate this study to them as an indication of their significance in this study as well as in my life.

Of course, I am deeply indebted to my father for his continuous prayer. I should not forget to express my thanks to my brothers and sisters for their help and support.

TABLE OF CONTENTS

ABSTRACT.......iii

ÖZ.......v

DEDICATION.......vii

ACKNOWLEDGMENTS.......viii

LIST OF TABLES.......xii

LIST OF FIGURES.......xiii

LIST OF SYMBOLS OR LIST OF ABBREVIATIONS.......xv

1 INTRODUCTION.......1

1.1 Background and Motivation.......1

1.2 Layout of the Thesis.......4

1.3 Contribution of the Thesis.......5

2 PRELIMINARIES.......6

2.1 Basic Definitions.......6

2.2 Symmetric Key Cryptosystems.......10

2.3 Asymmetric Key Cryptosystems.......11

2.4 Quality Encryption Measures.......11

2.4.1 Correlation Based Quality Measure.......11

2.4.2 Irregular Deviation Based Quality Measure.......12

3 HIll CIPHER AND ITS MODIFICATIONS LITERATURE SURVEY.......14

3.1 Original Hill Cipher.......14

3.1.1 Attacks.......16

3.2 Hill Cipher Modifications.......16

3.2.1 Hill Cipher Modification with Permutation Transfer HCM-PT 17

3.2.2 Hill Cipher Modification with the Number of Permutation Transfer HCM-NPT 19

3.2.3 Hill Cipher Modification with Hash Function HCM-H 20

3.2.4 Hill Cipher Modification with Hash-based Message Authentication Code HCM-HMAC 22

3.3 Conclusion 24

4 HILL CIPHER MODIFICATIONS BASED ON EIGENVALUES 25

4.1 Introduction 25

4.2 Hill Cipher Modification Based on Eigenvalues HCM-EE 25

4.3 Hill Cipher Modification Based on Pseudo-Random Eigenvalues HCM-PRE 27

4.4 Image Encryption Quality and Performance of the HCM-PRE and HCM-EE versus Known Ones 29

4.5. Security Analysis and Statistical Analysis 35

4.6 HCM-PRE and HCM-EE versus AES 41

4.7 Conclusion 43

5 COMMUTATIVE MATRIX-BASED DIFFIE-HELLMAN-LIKE KEY EXCHANGE PROTOCOL 44

5.1 Introduction 44

5.2 Overview of Key-Exchange Diffie-Hellman (DH) 46

5.3 The DH-Like Protocol 47

5.3.1 The Protocol 48

5.4 Security Analysis of the Protocol 49

5.5 Matrix-based DH-like Secure Protocol Extension 51

5.6 Conclusion 53

6 ElGAMAL PUBLIC KEY CRYPTOSYSTEM AND SIGNATURE SCHEME IN $GU(m,p,n)$ 55

6.1 Introduction 55

6.2 Overview of Original ElGamal Public Key Cryptosystem and Signature Scheme 56

6.3 Group $GU(m,p,n)$ and its Properties 56

6.4 ElGamal Public Key Cryptosystem and Signature Scheme in $GU(m,p,n)$ 59

6.4.1 Public key cryptosystem 59

6.4.2 Signature Scheme 60

6.5 Conclusion 62

7 CONCLUSIONS AND FUTURE RESEARCH 63

REFERENCES 65

APPENDIX 74

LIST OF TABLES

Table 2.1: The properties of modulo arithmetic………………………….………………7

Table 4.1: ID for encrypted images using HCM-PT, HCM-H, HCM-HMAC, HCM-EE and HCM-PRE, m=16…….……………………………………………………30

Table 4.2: Encryption time (msec) of Nike.bmp with HCM-PT, HCM-H, HCM-HMAC, HCM-EE and HCM-PRE.………………………………………………………31

Table 4.3: Correlation coefficients of two adjacent pixels in original and HCM-PT-encrypted images, HCM-H-encrypted images, HCM-HMAC-encrypted images………..…………………………………………………………….39

LIST OF FIGURES

Figure 2.1: Encryption and Decryption …..……………………………………..…….8

Figure 4.1: Nike.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE……………………………………………………………….32

Figure 4.2: Symbol.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE…………………………………………..………….32

Figure 4.3: Blackbox.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE……………………………………………………...33

Figure 4.4: Lena.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE……………………………………….................33

Figure 4.5: Girl.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE…………..…………………………………..................34

Figure 4.6: Encryption time (msec) of Nike.bmp with HCM-PT, HCM-H, HCM-HMAC, HCM-EE and HCM-PRE……………………………..……..……...34

Figure 4.7: Histogram of RGB layers for original/encrypted Nike.bmp: a) HCM-EE-encrypted, b) HCM-PRE-encrypted, c) histogram of the original image, d) histogram of   HCM-EE-encrypted e) histogram of HCM-PRE encrypted…….37

Figure 4.8: Histogram of RGB layers for original/encrypted Lena.bmp: a) HCM-EE-encrypted, b) HCM-PRE-encrypted, c) histogram of the original image, d) histogram of HCM-EE-encrypted e) histogram of HCM-PRE-encrypted ……..38

Figure 4.9. Correlation coefficients of two adjacent pixels in Nike.bmp encrypted by: HCM-EE, HCM-PRE, HCM-PT, HCM-H, and HCM-HMAC………………...40

Figure 4.10: Correlation coefficients of two adjacent pixels in Lena.bmp encrypted by: HCM-EE, HCM-PRE, HCM-PT, HCM-H, and HCM-HMAC………………...41

Figure 4.11: Mecy.bmp encrypted by: a) AES, b) HCM-EE, c) HCM-PRE…………….42

Figure 4.12: Bicycle.bmp encrypted by: a) AES, b) HCM-EE, c) HCM-PRE…………..42

Figure 4.13: Penguin.bmp encrypted by: a) AES, b) HCM-EE, c) HCM-PRE………….42

Figure 5.1: Computation of the session key K by MQV and HMQV (H is a hash function)……………………………………………………………………….52

Figure 5.2: Matrix-based analogue of HMQV…………………………………………...53

Figure Appendix 1: Main menu of the application……………………………………...76

Figure Appendix 2: Select an image to be encrypted…………………………………...77

Figure Appendix 3: Original image to be encrypted……………………………………77

Figure Appendix 4: Nike.bmp will be encrypted by HCM-PRE……………………….78

Figure Appendix 5: Nike.bmp HCM-PRE encrypted……..…………………….........78

Figure Appendix 6: Irregular deviation based quality measure for Nike.bmp…………78

Figure Appendix 7: Nike.bmp will be decrypted by HCM-PRE……………………….79

Figure Appendix 8: Nike.bmp HCM-PRE encrypted……………………..…………79

LIST OF SYMBOLS OR LIST OF ABBREVIATIONS

P: is the plaintext/plain-image fed as input to the encryption algorithm

C: is the ciphertext/cipher-image, the result of the encryption algorithm

C.C: denotes the correlation coefficient

ID: denotes the irregular deviation

E(x): denotes the overall mean value of x

h=histogram(DI): denotes the histogram distribution of DI

O: denotes the original/plain-image

DI: absolute value of the difference between each pixel value of the plain-image and the encrypted image

DC: average value of how many pixels are deviated at every deviation value

Z_m: the integer elements $Z_m=\{0,1,\ldots,m-1\}$

HC: Hill cipher

AES: Advanced encryption standard

DES: Data encryption standard

m: the block size

K: the key matrix

gcd: greatest common divisor

det(*K*): the determinant of the key matrix *K*

N: alphabet cardinality, *N*=256 for gray scale images

A: the sender

B: the receiver

K^{-1}: is the key inverse

KPCA: known plaintext ciphertext attack

HCM-PT: Hill cipher modification with permutation transfer

HCM-NPT: Hill cipher modification with the number of permutation transfer

HCM-H: Hill cipher modification with hash function

HCM-HMAC: Hill cipher modification with hash-based message authentication code

HILLMRIV: Hill cipher modification multiplying rows by initial vector

HCM-EE: Hill cipher modification based on eigenvalues

HCM-PRE: Hill cipher modification based on pseudo-random eigenvalue

NDK: the number of dynamic keys

K_t: the key matrix after permuting the rows and columns using permutation *t*

SEED: a seed value used to generate pseudo-random sequence permutation

t_r: PRPermutation(SEED,r), rth output permutation from the pseudo-random permutation generator, r is the block number

SHA: secure hash algorithm

MD5: Message-Digest Algorithm

$\lfloor x \rfloor$: the greatest integer that is less than or equal x

$\lceil x \rceil$: the nearest integer that is greater than or equal to x

DH: Diffie Hellman

HMQV: High performance Menezes, Qu and Vanstone protocol

GU(m,p,n): a group of numbers co-prime to mp^n, *p* is prime number and $m \neq 0$

$mo_{GF(p)}$ maximal order of *GF(p)*

U(mp^n): the group of units

Chapter 1

INTRODUCTION

1.1 Background and Motivation

The history of Cryptography can be tracked to the ancient civilizations in information secrecy and correspondence, such as ancient Egyptian civilization and the Romanian state. Cryptography algorithms are mathematical techniques inspired by the principles of basic mathematical, combination, permutation and logical operations which have add a number of security characteristics particularly useful for applications in engineering, and computer science, among other fields.

Nowadays, the term encryption has been commonly used to indicate hiding information. But the word "encryption" is imported from European languages it comes from the word “cipher”. Hence, the developments of use of the word cipher in almost all European languages to mean hiding information. Therefore, we can define cipher/encryption as hiding the information for its secrecy.

The recent advances in technology, especially in computer industry and communications, allowed potentially, enormous market for distributing digital information through the Internet. However, the proliferation of digital documents, multimedia processing tools, the worldwide availability of Internet access and network technologies have shown the urgent need of the presence of reliable security in storage and transmission of digital data. The security of multimedia data, digital speech data, images, as well as confidential video conferences is required in many applications since they are transmitted over open networks. Information security in general is provided by a method or a set of methods used to protect the data. These methods are heavily based on cryptography.

Cryptography has been intensively developed by researchers. The mathematician LESTER HILL in 1929 first invents the Hill cipher [1] [2], which marked the birth of modern cryptography. Cryptography is used to protect information to which illegal access is possible and where other protective measures are inefficient. The primitive operation of cryptography is encryption. It is a special computation that operates on messages; convert them into representation that is meaningless for all parties other than intended receiver.

In Cryptography, two classes of key-based encryption algorithms are used, symmetric (secret/private-key) and asymmetric (public-key); in symmetric algorithms same key is used for encryption and decryption (inverse of the key may be used for decryption) while asymmetric uses different keys for encryption and decryption.

The keys are considered as the basis for secure communication in modern cryptography, therefore, the process of creating (establishing) the secret keys is challenging problem for the symmetric cryptography to control encryption and decryption. Key establishment protocols provide shared secrets between two or more parties, typically for subsequent use as symmetric keys for a variety of information security services including encryption, message authentication, and entity authentication. One big issue with symmetric algorithm is the key exchange problem. However, the key exchange specifically is important from a security viewpoint, for each of the key-sharing parties can have its own control and a high confidence on the quality of the key output. In addition to encryption, key exchange is one of the notable problems in cryptography; it becomes another challenge in cryptography.

Considering the above points, in this thesis, the drawbacks of the Hill cipher algorithm and its known modifications has been studied, we proposed two new modifications of the original Hill cipher based on pseudo random eigenvalues [3][4], with the goal of generating dynamic encryption key efficiently to achieve high level of security. The Hill cipher is resistant to brute-force and statistical attacks, but it can be broken with a known plaintext-ciphertext attack (KPCA) [5].

A part of the thesis is devoted to the extension of the Diffie-Hellman key exchange protocol [6] and ElGamal cryptosystem [7].

The main contributions of the thesis are summarized as the following:

1. We propose two modifications of the Hill cipher, HCM-EE and HCM-PRE which are still resistant to brute-force and statistical attacks, and are resistant also to known plaintext-ciphertext attack (KPCA) due to dynamic encryption key matrix generating. With the modification, the new HCM-PRE can be applied widely in the systems which need high security (e.g., image encryption). Experimental results are given to demonstrate the proposed modifications that are significantly more effective in the encryption quality of images than original Hill cipher and its known modifications (HCM-PT, HCM-H, HCM-HMAC, and HCM-EE) in the case of images with large single colour areas, and slightly more effective otherwise.
2. A matrix-based Diffie-Hellman-like key exchange protocol and utilizing it as secure key-exchange protocol similar to HMQV are proposed. The proposed key exchange protocol uses matrix multiplication operation only; it does not rely on

the complexity of the discrete logarithm problem contrary to the prototype and its known variants. Two-way arrival at the common key, similar to that employed in the Diffie-Hellman protocol, is provided by specially constructed commutative matrices. The trap-door property ensuring the proposed protocol security is based on exploiting of a non-invertible public matrix in the key generating process.

3. ElGamal public key cryptosystem and signature scheme is extended to the group $GU(m,p,n)$ of numbers co-prime to mp^n and having analytical representation and known order. Elements of $GU(m,p,n)$ with the maximal order are used as the base elements in the proposed extension instead of primitive roots used in the original scheme. Proposed scheme allows easy periodic change of the group and base elements to provide necessary security level without change of the prime number p contrary to the case of $GF(p)$ used in the original ElGamal scheme. Computation of discrete logarithms in the proposed scheme is difficult for large p .

1.2 Layout of the Thesis

The rest of the thesis is divided into a number of chapters. Chapter 2 presents a brief introduction to cryptography concepts. Chapter 3 introduces a detailed literature survey of Hill cipher and its known modifications. Chapter 4 is devoted to the proposed Hill cipher modifications. Chapter 5 pauses to provide the necessary background for Diffie-Hellman key exchange protocol followed by a new cryptosystem consisting of the Diffie-Hellman-like key exchange matrix protocol. Chapter 6 is devoted to the extension of ElGamal public key cryptosystem and signature scheme to $GU(m,p,n)$. We conclude with some remarks in Chapter 7.

1.3 Contribution of the Thesis

The result of our research is summarized and reported in one journal paper and three conference papers that I finished during my PhD Studies.

1. In 2009, Hill Cipher Modification Based on Eigenvalues HCM-EE, Proc. of the Second International Conference on Security of Information and Networks (SIN2009) 6-10 October 2009, Gazimagusa (TRNC) North Cyprus, Elci, A., Orgun, M., and Chefranov, A. (Eds.) ACM, New York, USA, 2009: pp. 164-167.
2. In 2010, Secure Hill Cipher Modifications and Key Exchange Protocol, in Proc. 2010 IEEE International Conference on Automation, Quality and Testing, Robotics AQTR 2010- THETA 17th edition, Romania, Cluj-Napoca.
3. In 2010, ElGamal Public Key Cryptosystem and Signature Scheme in $GU(m, p, n)$, in Proc. 3rd International Conference on Security of Information and Networks 7-11 September 2010 Taganrog, Rostov-on Don, Russia
4. In 2011, Ahmed. Y. Mahmoud, Alexander. G. Chefranov, Hill Cipher Modification Based on Pseudo-Random Eigenvalues HCM-PRE to appear in the Journal of Applied Mathematics and Information Sciences (SCI-E)

Chapter 2

PRELIMINARIES

In this chapter, some basic concepts and definitions of cryptography are introduced.

2.1 Basic Definitions

In this section, we recall some standard mathematical notions and introduce some definitions from cryptography, which will be used throughout this work. Most of them can be found in [5].

The set of integers Z contains all integer numbers from negative infinity to positive infinity. The set of residues modulo N is Z_N . It contains integers from 0 to $N-1$. The set Z has non-negative (positive and zero) and negative integers; the set Z_N has only non-negative integers. To map a nonnegative integer from Z to Z_N , we need to divide the integer by N and use the remainder; to map a negative integer from Z to Z_N , we need to repeatedly add N to the integer to move it to the range 0 to $N-1$.

Modular Arithmetic: In the modular arithmetic system, the numbers are repeated after they reach a certain value (the modulus). If w, x , and y are three integers, N is positive integer and $Z_N = \{0,1,..,N-1\}$. The properties in Table 1 are held. The properties (Table 1) are valid for matrices that are residues of modulo arithmetic on a positive number N with entries over Z_N such that the matrix K satisfies (2.1)

$$gcd(det(K)\, mod\ N, N) = 1. \qquad (2.1)$$

where *det(K)* denotes the determinant of *K*, and *gcd* is the greatest common divisor.

Table 2.1: The properties of modulo arithmetic

Property	Expression
Identities	$(0+x)\, mod\ N = x\ mod\ N$ $(1 \cdot x)\, mod\ N = x\ mod\ N$
Commutative Law	$(x+w)\, mod\ N = (w+x)\, mod\ N$ $(x \cdot w)\, mod\ N = (w \cdot x)\, mod\ N$
Inverse	For each *w* belongs to Z_N , there exists *x* such that $(w+x)\, mod\ N = 0$ then $x = -y$ For each *w* belongs to Z_N and $gcd(w\ mod\ N, N) = 1$, there exists *e* such that $(w \cdot e)\, mod\ N = 1$, where *gcd* is the greatest common divisor
Associative Law	$[(y+x)+w]\, mod\ N = [y+(x+w)]\, mod\ N$
Distributive Law	$[w \cdot (x+y)]\, mod\ N = [w \cdot x + w \cdot y]\, mod\ N$ $[w \cdot (x \cdot y)]\, mod\ N = [(w \cdot x \bmod N) \cdot ((w \cdot y) \bmod N)]\, mod\ N$

All the matrices considered throughout the thesis are *m x m* sized with entries over Z_N , hence all the operations in encryption/decryption algorithms are assumed *mod N*, where *m* (block size) and *N* (alphabet cardinality) are selected positive integers (e.g., *N=256* for gray scale images). Also, we assume that two parties, *A* and *B*, want to communicate securely, and *A* is a sender, and *B* is a receiver.

Sender and Receiver: assume someone called the sender *A*, wants to send a message to a receiver, which we shall call the receiver *B*. Moreover, this sender *A* wants to send the message securely: s/he wants to make sure an eavesdropper/opponent cannot read the message.

Messages and Encryption: A message is a plaintext. The process of disguising a message in such a way as to hide its substance is encryption. An encrypted message is called ciphertext. The process of turning ciphertext back into plaintext is **decryption**. This is all shown in the Fig. 2.1:

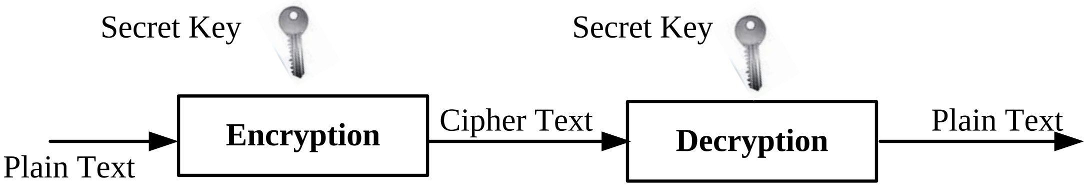


Figure 2.1: Encryption and Decryption

Plaintext: is denoted by *P*, for plaintext. This is the original message passed to the algorithm as input. It can be a stream of bits, a text file, a bitmap, digitized voice, a digital video image, etc.

Ciphertext: is denoted by *C*, for ciphertext. This is the encrypted plaintext produced as output of encryption algorithm. It depends on the plaintext and the used secret key. The encryptions of a given plaintext with two different keys yield two different ciphertexts. The ciphertext appears as random stream of data and, as it stands, unintelligible. The encryption process can be written as follows:

$$E_{k_e}(P) = C \tag{2.2}$$

where E is the **Encryption Algorithm** and k_e is used key for encryption; it performs various substitutions and transformations on the plaintext, P is the plaintext (original message) and C is the result of encryption algorithm (ciphertext). In the reverse process, the decryption D operates on C to produce the plaintext P, where k_d is key for decryption

$$D_{k_d}(C) = P \tag{2.3}$$

Since the whole point of encrypting and then decrypting a message is to recover the original plaintext, the following identity must hold true:

$$D_{k_d}(E_{k_e}(P)) = P \tag{2.4}$$

where k_d might be the same of k_e or its inverse in the case of symmetric encryption and k_d differ from k_e in the case of asymmetric encryption.

Secret key: The secret key is fed as input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

Decryption algorithm: This is essentially the reverse of encryption algorithm. The ciphertext and secret key are fed as input and produce the original plaintext.

Dynamic Keys: Dynamic keys are one-time symmetric cryptographic keys forming a sequence of keys. Every block in the plaintext is encrypted by a different cryptographic key. Instead of distributing the cryptographic keys among the parties, the dynamic keys are generated at participating parties. Unlike session keys which are exchanged among parties in every session, there is no key exchange at every session or transaction. A dynamic key generation scheme is used to produce a sequence of dynamic keys based on initial parameters. These parameters can either be pre-shared or exchanged via key exchange protocol only once at the beginning of the session. The number of distinct dynamic keys can be estimated based on the used initial parameters.

2.2 Symmetric Key Cryptosystems

All classical cryptosystems (cryptosystems that were developed before 1970s) are examples of symmetric key cryptosystems. In addition, most modern cryptosystems are symmetric as well. Some of the most popular examples of modern symmetric key cryptosystems include AES [8] (Advanced Encryption Standard), DES (Data Encryption Standard) [9] RC5 [10], Hill Cipher [1][2], and many others.

All symmetric key cryptosystems have a common property: they rely on a shared secret between communication parties. This secret key is used both as an encryption key and as decryption key (inverse of the key may be used for decryption). This type of cryptography ensures only confidentiality and fails to provide other objectives of cryptography. The important advantage over public (Asymmetric) key cryptosystems is that symmetric cryptosystems require smaller key sizes for the same level of security. Hence, the computations are much faster and the memory requirements are smaller. On the other hand the disadvantage of symmetric key cryptography is that it cannot handle large communication network of *n*-nodes needs to communicate with confidentially with all other nodes in the network, it needs *n-1* shared secrets. For large value of *n* this is highly impractical and inconvenient. To overcome this disadvantage, the key exchange protocols can be used to exchange the keys between the parties.

2.3 Asymmetric Key Cryptosystems

In asymmetric key cryptosystems there are two different keys: a public key, which is publicly known, and the secret key, which is kept secret by the owner. The system is called “Asymmetric” since the different keys are used for encryption and decryption, the public key and the private key.

If data is encrypted with a public key, it can be decrypted only by using the corresponding private key. Today, all public key cryptosystems rely on some computationally difficult problems. For example, the cryptosystem RSA [11] relies on difficulty of factoring large integers, while El-Gamal [12] cryptosystem relies on discrete logarithm problem DLP of a group element with generator base in finite Abelian group.

2.4 Quality Encryption Measures

A number of different evaluation measures have been used to measure the encryption quality of images/signals. The most widely used and popular measures are correlation coefficients (C.C) and irregular deviation based quality (ID) [13][14][15]. In this section we recall C.C and ID which will be used to measure the image encryption quality.

2.4.1 Correlation Based Quality Measure

A good encryption algorithm must produce an encrypted image of totally random patterns hiding all the features of the original image, and the encrypted image must be independent of the original image. This means that the two images must have a correlation coefficient very close to zero. The correlation coefficient is given by the following expression:

$$C.C = \frac{\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{N}(x_i - E(x))^2}\sqrt{\sum_{i=1}^{N}(y_i - E(y))^2}}. \tag{2.5}$$

where x_i and y_i are the data value of plain-image/signal and encrypted-image/signal at point i, respectively, and E denotes the overall mean value. The closer C.C to zero, the better.

2.4.2 Irregular Deviation Based Quality Measure

This quality measuring factor is based on how much the deviation affected by encryption is irregular. This quality measure can be formulated as follows:

1. Calculate the matrix, *DI*, which represents the absolute value of the difference between each pixel value of the original/plain-image and the encrypted image respectively:

$$DI = |O - E|, \tag{2.6}$$

where *O* is the original (input) image and *E* is the encrypted (output) image.

2. Construct a histogram distribution of the *DI* we get from step 1:

$$h{=}histogram\ (DI). \tag{2.7}$$

3. Get the average value of how many pixels are deviated at every deviation value by:

$$DC = \frac{1}{256}\sum_{i=0}^{255} h_i , \tag{2.8}$$

4. Subtract this average from the deviation histogram and take the absolute value

by:

$$AC(i) = |h_i - DC|. \tag{2.9}$$

5. Count:

$$ID = \sum_{i=0}^{255} AC(i). \tag{2.10}$$

The smaller ID, the better.

Chapter 3

HILL CIPHER AND ITS MODIFICATIONS

LITERATURE SURVEY

Letter-by-letter substitution ciphers are not resistant against frequency analysis and so notoriously unsecure. In a block cipher the plaintext is divided into groups of adjacent letters of the same fixed length *m*, and then each such group is transformed (encrypted) into a different group of *m* letters according to some key instead of substituting letters individually. If *m* is large enough, it can be more challenging to break and can resist the frequency analysis. The first systematic simple block cipher using more than two letters per group is the Hill cipher. Hill cipher is invented by the mathematician Lester Hill [1][2].

3.1 Original Hill Cipher

The Hill cipher (HC) is one of the notoriously symmetric cryptosystem. The main operation of HC is matrix manipulations; it multiplies a plaintext vector by a key matrix to get the ciphertext. It is very attractive due to its simplicity and high throughput [16][17].

The basic idea of the HC is to put the letters of the plaintext into blocks of length *m*, assuming an *m x m* key matrix, and then each block of plaintext letters is then converted into a vector of integers according to the alphabet chosen and then multiplied by the *m x m* key matrix. The results are then converted back to letters and the ciphertext message is produced. The key for HC system consist of an *m x m* square invertible matrix K , where the larger the dimensions the more secure the encryption will be. To ensure the key matrix K is invertible, the *det(K)* must be relatively prime to the modulus *N*, to satisfy this we require

$$gcd(det(K)\,mod\;N\;,N\;)=1\,. \tag{3.1}$$

where *m* (block size) and *N* (alphabet cardinality) are selected positive integers (e.g., *N=256* for gray scale images), *det(k)* the determinant of *K* and *gcd* is the greatest common divisor. The HC has the property of diffusion: when one changes one letter in the plaintext, several letters of the ciphertext are changed. This makes it much more difficult to use frequency tests. It also has the property of confusion: each letter of the ciphertext depends of several parts of the key. Then the key cannot be computed part by part.

Suppose two parties, a sender, *A*, and a receiver, *B*, want to exchange data using HC; they share securely a non-singular invertible key matrix *K*. If *A* wants to encrypt a plaintext vector, *P*, he gets the ciphertext vector, *C*, as follows:

$$C = K \cdot P \bmod N \;. \tag{3.2}$$

The receiver, *B*, decrypts the ciphertext vector *C* by

$$P = K^{-1} \cdot C \bmod N \;. \tag{3.3}$$

where K^{-1} is the key inverse and *N* is the alphabet cardinality. For existence of K^{-1}, we require to satisfy (3.1).

3.1.1 Attacks

The HC is extremely secure (resistant) against ciphertext only and brute force attacks. That is because the key space is very large, due to choosing the matrix elements from a large set of integers [17], it is also resistant to the frequency letter analysis, and statistical analysis while it can be broken with a known plaintext-ciphertext attack (KPCA) [5]. The key matrix can be calculated easily from a set of known plaintext and ciphertext pairs. The KPCA works as follows:

Suppose that opponent has "captured" enough plaintext along with the corresponding ciphertext, and he/she constructs m blocks of m letters of plaintext. Write each block as a vertical vector P_i $(1 \le i \le m)$ and each block of corresponding ciphertext as a vectorC_i. Then, for each i the opponent has: $K \cdot P_i = C_i$, where K is the unknown key matrix. Form a $m \times m$ matrix P with the m vertical vectors of plaintext as columns $P = [P_1 ... P_m]$ and similarly, form a $m \times m$ matrix C with the m vertical vectors of ciphertext as columns$C = [C_1 ... C_m]$. Then $K \cdot P = C$. If P is invertible $mod\ N$, then we can find$K = C \cdot P^{-1}$. If P is not invertible $mod\ N$, we can try to find other blocks of plain text. Once you have computed the key for the HC, then of course the opponent can reveal all the plaintext enciphered by that key and he might impersonate the sender and cheat (deceive) the receiver by using the key to create fake ciphered messages to send them.

3.2 Hill Cipher Modifications

Most of the Hill cipher modifications were developed in the last two decades [15] [16][18][19][20]. The aim of those modifications was to repair the weaknesses of the HC, due to its succumbed to a KPCA.

However most of these modifications were oriented and tested for image encryption. Image encryption has large applications in internet communications; it is widely used in multimedia systems. It is shown in the literatures that almost all the previous modifications of HC are either insecure or not effective for image encryption [3][4][6][18][20][21][22].

3.2.1 Hill Cipher Modification with Permutation Transfer HCM-PT

HC modification [16], HCM-PT, uses a dynamic key matrix obtained by random permutations of rows and columns from the master key matrix to get every next ciphertext, and transfers it together with an HC-encrypted permutation to the receiving side. Thus, in HCM-PT, each plaintext vector is encrypted by a new dynamic key matrix that prevents the KPCA on the vectors. The number of possible dynamic keys is equal to the number of permutations of the key matrix rows, and it may be used as a characteristic of its security. But permutations in HCM-PT are transferred HC-encrypted, which means that master key matrix can be revealed by the KPCA on the transferred encrypted permutations [18].

The HCM-PT differs from (3.2), (3.3) as follows: To encrypt a plaintext P , A selects a permutation, t , randomly over Z_m , builds a permutation matrix M_t , by pre-agreed way, where each row and column of which has all zero entries except only one non zero entry equal to one only, and gets K_t by permuting the rows and columns of a key matrix K getting

$$K_t = M_t \, K \, M_t^{-1} \, . \tag{3.4}$$

The HCM-PT encryption is then performed by (3.1), but using K_t instead of K .

Additionally, sender A encrypts t by (3.2) using K and getting u as a ciphertext, and sends C and u together to the receiver.

In order to decrypt the ciphertext, B decrypts t from u by using (3.3), gets $(K^{-1})_t = (K_t)^{-1}$ [16] from K^{-1}, and then reveals the plaintext by (3.3), using $(K^{-1})_t$ instead of K^{-1}. The number of dynamic keys used in HCM-PT is

$$NDK(HCM - PT) = m!, \tag{3.5}$$

3.2.1.1 Attack

The HCM-PT is resistant against the attacks, which resisted by HC. But HCM-PT, can be broken with KPCA [18] due to permutations in HCM-PT are transferred HC-encrypted. The KPCA can be applied on HCM-PT as follows:

The permutations are transferred HC-encrypted as $u = K \cdot t \bmod m$, this is exactly the same problem as the original Hill cipher. Suppose the opponent collected m pairs of (u, t), the opponent (cryptanalyst) can reveal the key K. However, the opponent can obtain the permutation matrix M_t associated to t. Hence, the opponent can calculate the key matrix K_t by (3.4).

On the other hand, if the permutation t cannot be obtained, suppose that the opponent has captured m pairs of plaintext with the corresponding ciphertext (C, P) to get K_t. It is known that, the ciphertext has been obtained by applying (3.2) using K_t instead of K $(C = K_t \cdot P \bmod N)$, in addition the opponent knows that $u = K \cdot t \bmod m$ and K_t is calculated by (3.4). Therefore, from the former scenario, the opponent can obtain the following [18]:

$$K = [U][T]^{-1}, \tag{3.6}$$

where

$$U = [u_1 u_2 .. u_m]$$

and (3.7)

$$T = [t_1 t_2 .. t_m]$$

are $m \times m$ matrices $K_t = M_t \, K \, M_t^{-1} \Leftrightarrow K = M_t^{-1} K_t \, M_t$,

From (3.6) and (3.7), the equations can be rewritten as

$$\begin{aligned}
&[U][T]^{-1} = M_{t_1}^{-1} K_{t_1} M_{t_1} \\
&[U][T]^{-1} = M_{t_2}^{-1} K_{t_2} M_{t_2} \\
&\vdots \\
&[U][T]^{-1} = M_{t_m}^{-1} K_{t_m} M_{t_m}
\end{aligned}$$

Suppose that the predefined function $t \Rightarrow M_t$ is known and $[T]^{-1}$exists, and then the permutation *t* is obtained by solving the *m* equations. This means that, the opponent can collect *m* pairs of the parameters to solve the equations $[U][T]^{-1} = M_{t_m}^{-1} K_{t_m} M_{t_m}$ and *m* parameters to reveal (calculate) each K_t from$C = K_t \cdot P \bmod N$. Finally, the key *K* can be obtained by m^2 known-plaintext pairs (u, P, C)

3.2.2 Hill Cipher Modification with the Number of Permutation Transfer HCM-NPT

The HCM-NPT [19] cipher is a modification of HCM-PT which, in turn, is a modification of HC. HCM-NPT uses the same initialization and the same encryption/decryption technique as HCM-PT does, but without permutations transfer; instead, both communicating parties use a pseudo-random permutation generator, and only the consecutive number of the necessary permutation is transferred to the receiver. It has good computational complexity and the number of its dynamic keys is the same as for

HCM-PT. HCM-NPT assumes that the sender, *A*, and the receiver, *B*, share a secret seed value, *SEED*, which is used to generate a pseudo-random sequence of permutations.

In order to encrypt a plaintext, the sender, *A*, selects a number *r*, and calculates

$$t_r = PRPermutationG(SEED, r), \tag{3.8}$$

getting the *r*-th output permutation from the pseudo-random permutation generator *PRPermutationG* (*r* can be a block number in the sequence of transmitted blocks, or its function). Sender *A* then gets a ciphertext *C* as in HCM-PT, and sends to receiver *B* both *C* and *r*. In order to decrypt, *B* calculates t_r according to (3.8), and then gets the plaintext as in HCM-PT. The number of dynamic keys used in HCM-NPT, *NDK(HCM-NPT)*, is the same as *NDK(HCM-PT)* (3.5). It is shown in [3][4][5], neither HCM-NPT nor HCM-PT are effective for image encryption with images containing very large single colour areas.

3.2.3 Hill Cipher Modification with Hash Function HCM-H

HC modification [18], HCM-H, also uses dynamic key matrix produced with the help of a one way hash function applied to an integer picked up randomly by the sender to get the key matrix, and a vector added to the product of the key matrix with a plaintext. HCM-H is computationally expensive due to the use of hash function. On the other hand, it was assumed that HCM-H solved the drawbacks in the original HC and is secure, but recently, it is proved that HCM-H is vulnerable [20] to chosen-ciphertext attack because the selected random number is transmitted in clear over the communication link and is repeated.

HCM-H, works as follows. The sender, *A*, and the receiver, *B*, share an invertible matrix *K*. To encrypt the plaintext P , *A*, selects a random integer *a*, where $0 < a < N$, and applies a one way hash function to compute the parameter $b = f(a \| k_{11} \| k_{12} \| \ldots \| k_{mm})$, where $k_{11}, k_{12}, ..., k_{mm}$ are the elements of K ; *b* is used to select the k_{ij} from *K*, where *i* and *j* can be calculated according to (3.9)

$$i = \left\lfloor \frac{b-1}{m} \right\rfloor \cdot (\bmod m) + 1,\ j = b - \left\lfloor \frac{b-1}{m} \right\rfloor \cdot m. \tag{3.9}$$

Then, *A* generates a vector $V = [v_1, v_2, ..., v_m]$ according to (3.10)

$$\begin{aligned} &v_1 = f(k_{ij}) \bmod N, \\ &v_2 = f(v_1) \bmod N = f^2(k_{ij}) \bmod N, \\ &..., \\ &v_m = f(v_{m-1}) \bmod N = f^m(k_{ij}) \bmod N. \end{aligned} \tag{3.10}$$

Then, *A* encrypts the plaintext *P* by

$$C = k_{ij} \cdot P \cdot K + V \bmod N, \tag{3.11}$$

and sends together *C* and a to *B*. The decryption process is done by

$$P = k_{ij}^{-1} \cdot (C - V) \cdot K^{-1} \bmod N. \tag{3.12}$$

The number of dynamic keys used in HCM-H is

$$NDK(HCM\text{-}H) = min(m^2, N). \tag{3.13}$$

3.2.3.1 Attack

The encryption of HCM-H can be done using (3.11). The encryption of the *t*-th block plaintext P_t can be done by (3.11) which has the form $C_t = Y_t \cdot P_t \cdot K + V_t \bmod N$, where Y_t is the corresponding k_{ij} . It is shown in [18], the KPCA cannot applied on HCM-H even if the opponent knows *m* pairs of $(P_t, C_t),\ 1 \le t \le m$, due to the key matrix and parameters Y_t and V_t are unknown and m equations cannot be used for solving an unknown $m \times m$ matrix and $2m$ unknown parameters. But, in [20] it is shown that HCM-H is vulnerable to the chosen-ciphertext attack in the case the opponent selects those equations have the same Y_t and V_t .

The chosen-ciphertext attack works as follows:

The opponent selects different ciphertexts in which he has access to the corresponding plaintexts. The opponent tries to w, the reveal the key. The chosen ciphertext attack is most relevant to the public-key algorithms; it also can be used effectively against the symmetric algorithms.

The weakness of HCM-H against the chosen ciphertext attack due to the values of *b* and *V*, and the selection of k_{ij} depend on the value of *a*, and their values don't differ for the same value of *a*. The value of *a* is selected randomly but it is sent in clear form over the communication which allow the opponent (eavesdropper) to easily capture and use it for chosen-ciphertext attack. The chosen-ciphertext attack can be applied on HCM-H as follows:

Suppose that the sender *A*, sends the pairs (C, a) to the receiver *B*, The opponent eavesdrops, capture and saves them. The random number will be repeated soon or later in some pairs $(C\ ,\ a)$. The opponent selects $(m+1)$ pairs of $(C\ ,\ a)$`that have the same random number a . Based on the chosen-ciphertext attack, the opponent has access to the corresponding plaintext for the chosen ciphertexts. The opponent has a set of equations$C_t = k_{ij} \cdot P_t \cdot K + V_t \bmod N$, $1 \le t \le m+1$where P_t and $C_t$are known parameters. The opponent can easily obtain (reveal) the key matrix *K*. The vector *V* can be easily eliminated from pairs encrypted with the same random number.

3.2.4 Hill Cipher Modification with Hash-based Message Authentication Code HCM-HMAC

The HCM-HMAC [20] cipher is a modification of HCM-H, the aim of HCM-HMAC is to avoid the random number transfer in HCM-H. It uses only a seed value secure transfer, and then both parties generate necessary numbers synchronously, where HMAC is a hash function, e.g., SHA-1[5], MD5 [23]. The difference between HCM-H and HCM-HMAC is similar to the difference between HCM-PT and HCM-NPT.

The HCM-HMAC, works as follows. In order to transfer a seed value, the sender, *A*, transmits the seed value a according to the Hughes key-exchange protocol [24]. Then the seed value a_0 can be used to generate the chain of pseudo-random numbers synchronously by the both parties; a_t can be calculated by

$$a_t = HMAC_{k'}(a_{t-1}), t = 1, 2, \ldots, \qquad (3.\ 14)$$

where k' is the secret key of the hash function, k' can be calculated by

$$k' = (k_{11} \| k_{12} \| k_{13} \| \ldots \| k_{mm} \| a_{t-1}) \bmod 2^q, \qquad (3.\ 15)$$

where || denotes the concatenation, *q* is the number of bits required for the hash function, and a_t is used in recursive calculations of the vector $V=[v_1, v_2, \ldots, v_n]$, calculated for the encryption of *t-th* block, $v_0 = 1$, if $a_t \equiv 0 (\bmod p)$ otherwise $v_0 = a_t \bmod p$, *p* is a prime number.

$$v_i = k_{ij} + \tilde{v}_{i-1} a_t \bmod p, i = 1, 2, \ldots m, \text{ and } j = (v_{i-1} \bmod m) + 1 \qquad (3.16)$$

$\tilde{v}_{i-1}$ is calculated by

$$\tilde{v}_{i-1} = 2^{\left\lceil \frac{\gamma}{2} \right\rceil} + \left(v_{i-1} \bmod 2^{\left\lceil \frac{\gamma}{2} \right\rceil} \right), \qquad (3.17)$$

where $\gamma = \lfloor \log_2 v_{i-1} \rfloor + 1$ denotes the bit length of v_{i-1}. Then, *A* encrypts the plaintext P_t by

$$C_t = v_0 \cdot P_t \cdot K + V \bmod p, \qquad (3.18)$$

and sends together C_t and a to B, t=1,2,... The receiver B calculates the required parameters by using (3.12)-(3.16), and then gets the plaintext by

$$P_t = v_0^{-1} \cdot (C_t - V) \cdot K^{-1} \bmod p \text{ .} \tag{3.19}$$

3.3 Conclusion

The Hill cipher is very attractive due to its simplicity and high throughput [16][17]. Its attributes including its cryptanalysis are reported in some cryptographic textbooks [5][24][25][26]. The vulnerability of the HC and its weaknesses against the KPCA make it unusable in practice. Although several HC modifications have been proposed to improve the security of the HC, but the proposed HC modifications either still susceptible, vulnerable to the cryptanalytic attacks and have the same essential drawbacks of the original HC or they are not effective for encryption of images with large single colour areas. A challenging problem is to improve the security of HC/HC-modifications and make it effective for image encryption since neither HC nor known HC-modifications are effective for image encryption in large area with single colour.

Chapter 4

HILL CIPHER MODIFICATIONS BASED ON EIGENVALUES

4.1 Introduction

In this chapter, we present our proposed modifications [4] of the Hill cipher, HCM-EE, generating dynamic encryption key matrix efficiently with the help of eigenvalues [27], it uses the eigenvalues for matrix exponentiation to a pseudo-random power for a new key matrix generated for each plaintext block. The proposed approach for improving the Hill cipher security is presented in section 4.2. Section 4.3 includes another modification of HC, HCM-PRE [3], based on the use of pseudo-random eigenvalues to construct a key matrix [27] and modify it for each new plaintext. In order to verify the importance of the resultant observations from encryption quality viewpoint, the results of the conducted experiments are shown in section 4.4. The security and statistical analysis are presented in section 4.5. Section 4.6 shows encryption quality of images encrypted by HCM-EE and HCM-PRE versus AES. Finally, we conclude with some notes in section 4.7.

4.2 Hill Cipher Modification Based on Eigenvalues HCM-EE

In [4] we propose a modification of Hill cipher denoted as HCM-EE; HCM-EE works as follows. Sender *A* selects a set$E=\{e_1,e_2,\ldots,e_m\}\subset Z_N-\{0\}$, *gcd*($e_j$, *N*)=1, *gcd* is the greatest common divisor, $1\leq j\leq m$; at least one e_j should have the maximal order which is $\frac{\varphi(N)}{2}$ for

N being a power of 2 [28], $\varphi(N)$ is the Euler's totient function [5], giving the number of positive integers less than *N* and co-prime to it. Then *A* constructs an invertible matrix *Q* and calculates the key matrix *K* [27]:

$$K = Q \cdot D \cdot Q^{-1}, \tag{4.1}$$

where D is a diagonal matrix, diagonal elements of which are its eigenvalues from *E*. Note that *Q* and *D* satisfy (3.1); *A* and *B* share them securely. Additionally, they share the secret values, *SEEDl* and *SEEDt*; *SEEDl* is used to generate the set of pseudo-random numbers $l = \{l_1, l_2, ..., l_n\}$ by (4.2), $l_i \neq 0$ and $l_i \in \{2, ..., \varphi(N) - 1\}$, $1 \leq i \leq n$, n is the number of blocks. *SEEDt* is used to generate a pseudo-random sequence of permutations *t*. In order to encrypt the *i*-th plaintext block P_i, *A* selects

$$l_i = PRNG(SEEDl, i) > 0, \tag{4.2}$$

then calculates

$$E_i = \{e_j^{l_i}\}_{t_r}, 1 \leq j \leq m, 1 \leq i \leq n, \tag{4.3}$$

where $e_j \in E$, n is the number of blocks, and the random permutation t_r can be obtained by (3.8). Finally, *A* calculates

$$K_i = Q \cdot D_i \cdot Q^{-1}, \tag{4.4}$$

where D_i is a diagonal matrix, diagonal elements of which are from E_i and

$$i = \frac{\varphi(N)}{2} \cdot r + s, \; 0 \leq s < \frac{\varphi(N)}{2} \; . \tag{4.5}$$

The plaintext P_i is encrypted as follows

$$C_i = K_i \cdot P_i + diag(D_i), \tag{4.6}$$

where $diag(D_i)$ is a vector of the main diagonal elements of D_i.

In order to decrypt the ciphertext, *B* computes l_i according to (4.2), t_r according to (3.8) and (4.5), E_i according to (4.3), and

$$(K_i)^{-1} = (Q \cdot D_i \cdot Q^{-1})^{-1} = Q \cdot D_i^{-1} \cdot Q^{-1}. \qquad (4.7)$$

Then, *B* retrieves the plaintext:

$$P_i = K_i^{-1} \cdot (C_i - diag\,(D_i)). \qquad (4.8)$$

It is appropriate to mention that for computing K_i we use a diagonal matrix, and only the diagonal entries of D_i are exponentiated to the power l_i, requiring $O(\,mlog_2 l_i\,)$ multiplications. On the other hand, to get D_i^{-1}, we calculate the inverse of *m* numbers only. Note also that Q^{-1} and D_i^{-1} are calculated only once. The diagonal elements of D_i^{-1} belong to the group G of numbers co-prime to *N*. Based on Theorem 10.3 [28] we see that for *N*=256, $64 = \frac{\varphi(N)}{2}$ is the maximal order of elements of G (odd numbers in Z_{256}). In HCM-EE, we select at least one element in the diagonal with the maximum order to guarantee the maximum period of the diagonal elements. The number of dynamic keys of HCM-EE is estimated as

$$LB \cdot m! \leq NDK(HCM\text{-}EE) \leq \varphi(N) \cdot m!. \qquad (4.9)$$

where LB is the maximum order of the diagonal elements in D_i. If N is a power of 2, $LB = \frac{\varphi(N)}{2}$. It is assumed that the sender and receiver will exchange all the shared parameters by using the proposed protocol [6].

4.3 Hill Cipher Modification Based on Pseudo-Random Eigenvalues HCM-PRE

We propose another Hill Cipher Modification based on Pseudo-Random Eigenvalues;

denoted as HCM-PRE [3]. The proposed HCM-PRE uses the same encryption/decryption technique as HCM-EE [4] does. But HCM-PRE differs from the HCM-EE in the key construction. It uses pseudo-random eigenvalues instead of static eigenvalues exponentiated to pseudo-random powers in HCM-EE. Similar to HCM-EE, HCM-PRE assumes the sender, *A*, and the receiver, *B*, uses the proposed protocols in [6] to exchange securely all the secret parameters.

If the sender, *A*, and the receiver, *B*, want to communicate using HCM-PRE, they share a secret value, SEED, that is used to generate pseudo-randomly a sequence of eigenvalue sets, $E = (E_i), 1 \le i \le n$:

$$E = PRSetG_{SEED}(n, m) , \tag{4.10}$$

where $E_i = \{e_{ij}\} \subset Z_N - \{0\}$ is a set of eigenvalues of the matrix to be constructed, e_{ij} is relatively prime to N , $1 \le j \le m, 1 \le i \le n$, for positive integers n and m , n is the number of blocks; $PRSetG_{SEED}(n, m)$ is a pseudo-random set sequence generator (using e.g., RC4 initialized by *SEED*) returning n sets, each of which contains m numbers. Sender *A* then constructs an invertible matrix Q as in HCM-EE. The key matrix K_i is calculated by (4.4) but, instead of static eigenvalues used in the diagonal elements of D_i , diagonal matrix D_i is used, diagonal elements of which are all the eigenvalues from E_i , $1 \le i \le n$. HCM-PRE uses a different set of diagonal elements for every plaintext. It may be easily shown that K_i is invertible modulo N since Q and D_i have (by construction) determinants relatively prime to N . Finally, the plaintext P_i is enciphered by (4.6).

To decrypt a ciphertext, receiver *B* computes E according to (4.10), and finds K_i^{-1} using (4.7). Note that to get D_i^{-1}, we calculate the inverse of m numbers only, and that

Q and Q^{-1} are constructed only once. Receiver B then retrieves the plaintext by (4.8). To generate an invertible key matrix D_i, the eigenvalues must be in the multiplicative group of Z_N, the number of possible eigenvalues in the multiplicative group of Z_N is $\varphi(N)$. Hence the number of dynamic keys of HCM-PRE is

$$NDK(HCM-PRE) = \min\left(\varphi(N)^m, \frac{Period(RC4)}{m}\right) \tag{4.11}$$

where *Period(RC4)* is overwhelmingly likely to be greater than 10^{100} [29].

4.4 Image Encryption Quality and Performance of the HCM-PRE and HCM-EE versus Known Ones

The experiments are hosted on a Windows XP OS running on a Dell Latitude D630 laptop with Intel(R) Core(TM) 2 Duo 1.8 GHz processor and with 2-GB RAM. The simulation is implemented by Visual Studio Environment version 2008. The performance evaluation tool used is as C# application, which provides a wide range of profiling instruments for reading and manipulating images (a brief description of the application is given in the appendix). In our experiments, several RGB images are encrypted. Firstly, the image, P, of size NxM is converted into its RGB components. Afterwards, each colour matrix *(R, G, B)* is converted into a vector of integers within $\{0,1,...,255\}$. Each vector has the length $L = NxM$. Then, the so obtained three vectors represent the plaintext $P(3\times L)$ which will be encrypted using the block size *m*=16.

We examine the encryption quality for three different images containing very large single colour areas: Nike.bmp (Fig. 4.1), Symbol.bmp (Fig. 4.2), and Blackbox.bmp (Fig. 4.3). Also we examined the encryption quality for an image that does not contain many high frequency components: Lena.bmp (Fig. 4.4). The Girl.bmp (Fig. 4.5) is used as an example of an image containing many high frequency components. Each image is encrypted using HCM-PT, HCM-H, HCM-HMAC, HCM-EE, and HCM-PRE.

The quality of encryption of these images is studied by visual inspection (Figs. 4.1-4.5) and quantitatively (Table 4.1, used irregular deviation based quality measure ID [12][13][16] is explained in Chapter 2).

Table 4.1: ID for images encrypted by HCM-PT, HCM-H, HCM-HMAC, HCM-EE and HCM-PRE, m=16.

Image/Algorithm	HCM-PT	HCM-H	HCM-HMAC	HCM-EE	HCM-PRE
Nike.bmp	23980.79	13171.75	9983.87	2656.62	1338.04
Symbol.bmp	10482.25	5755.68	4830.91	2378.07	1874.30
Blackbox.bmp	34036.28	18511.62	11491.48	3285.25	1328.63
Lena.bmp	10256	10518.66	10469.33	10172.66	10201.33
Girl.bmp	11459.55	10472.61	10336.77	9942.21	9913.25

Based on visual inspection, it is obvious that the HCM-PRE and HCM-EE are better than the HCM-PT, HCM-H, and HCM-HMAC in hiding all the features of the image containing large single colour areas (Figs. 4.1-4.3).

Based on the numerical evaluation of encryption quality measure ID (Table 4.1), we note that the proposed scheme HCM-PRE versus HCM-EE give alternately better or nearly the same encryption quality. Table 4.1 shows also that the proposed scheme; HCM-PRE is more effective in encryption quality than HCM-PT, HCM-H, HCM-HMAC, and HCM-EE. On the other hand, HCM-PT, HCM-H, HCM-HMAC, HCM-EE, and HCM-PRE are all good in encrypting images containing many high frequency components; all the algorithms give nearly the same results but the HCM-PRE and HCM-EE are the most effective ones (Table 4.1, rows 4-5).

We examined the encryption time for the Nike.bmp image having $124x124$ pixels and 45KB size. The encryption time measured when applying HCM-PT, HCM-H, HCM-HMAC, HCM-EE, and HCM-PRE is shown in (Table 4.2 and Fig. 4.6). In our implementation, HCM-EE and HCM-PRE were used with RC4 [5] for the pseudo-random permutation generator (3.6), pseudo-random number generator (4.2) for HCM-EE, and pseudo-random set generator (4.9) for HCM-PRE. We implemented HCM-H with SHA-1 [5] since the latter has been used in [18], and the built-in HMAC from C# with HCM-HMAC-SHA-1. Table 4.2 and Fig. 4.6 show that HCM-PRE has the best execution time; it is roughly two times faster than HCM-EE and HCM-HMAC, and four times faster than HCM-H. HCM-EE roughly is twice better than HCM-H and it has nearly the same execution time as of HCM-HMAC but HCM-EE has better encryption quality (Figs. 4.1-4.5, and Table 4.1). Table 4.2 shows that HCM-PT is faster than HCM-EE but equations (3.5) and (4.8) show that NDK(HCM-EE) is greater than NDK(HCM-PT), hence HCM-EE is more secure than HCM-PT. Equation (4.11) shows that NDK(HCM-PRE) is greater than NDK(HCM-EE). Hence HCM-PRE is more secure and is more effective in the encryption time than HCM-PT, HCM-H, HCM-HMAC and HCM-EE.

Table 4.2: Encryption time (msec) of Nike.bmp with HCM-PT, HCM-H, HCM-HMAC, HCM-EE and HCM-PRE.

HCM-NPT	HCM-H	HCM-HMAC$_k$	HCM-EE	HCM-PRE
103	425	214	200	98

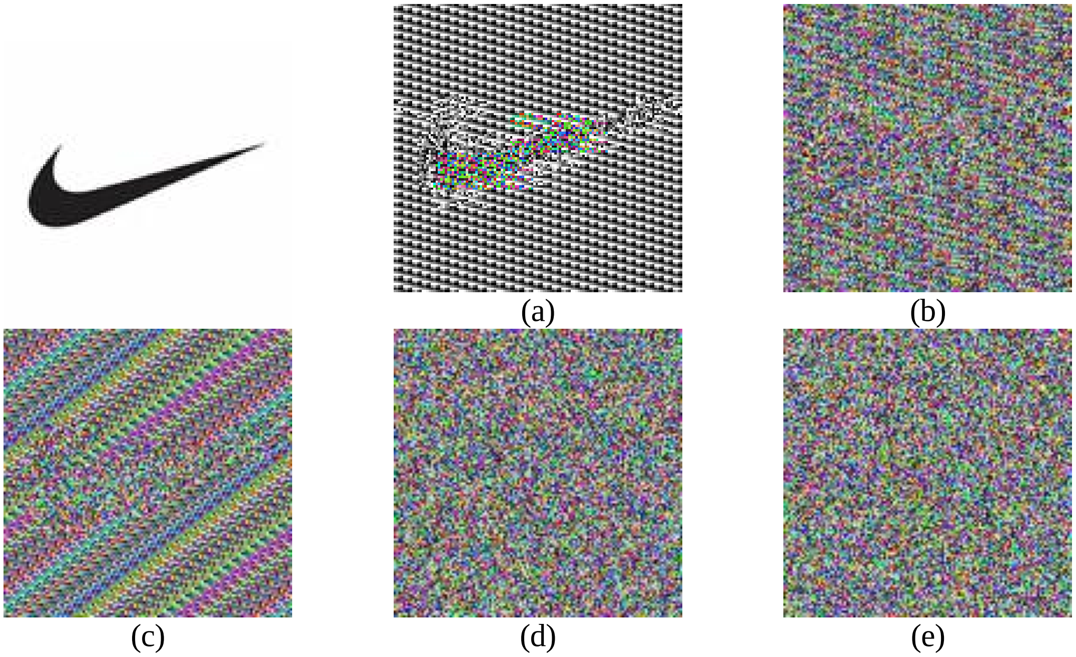


Figure 4.1: Nike.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE.

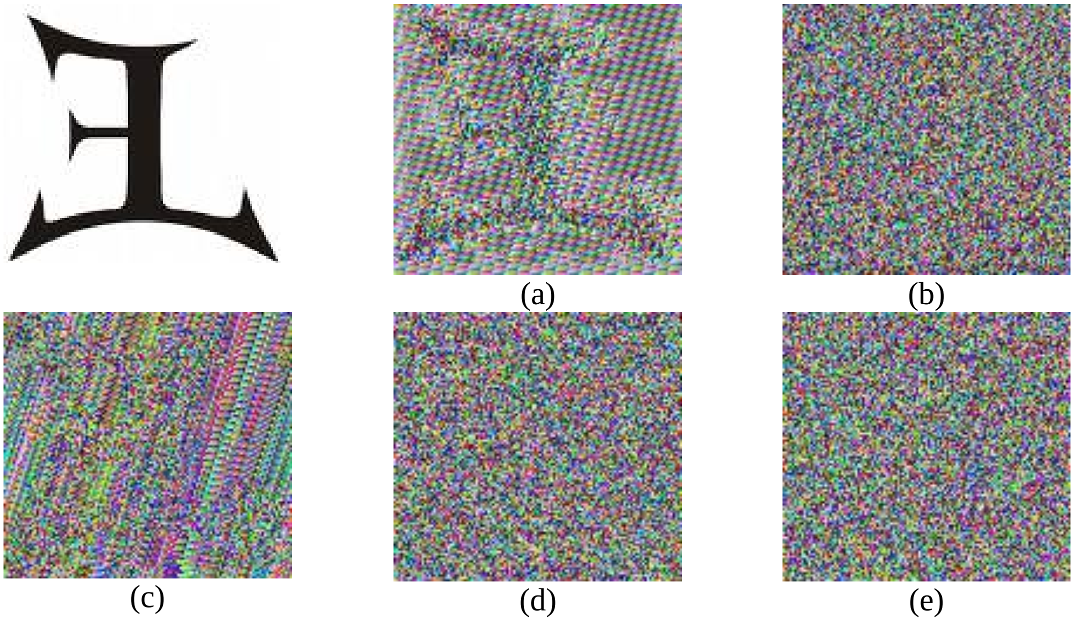
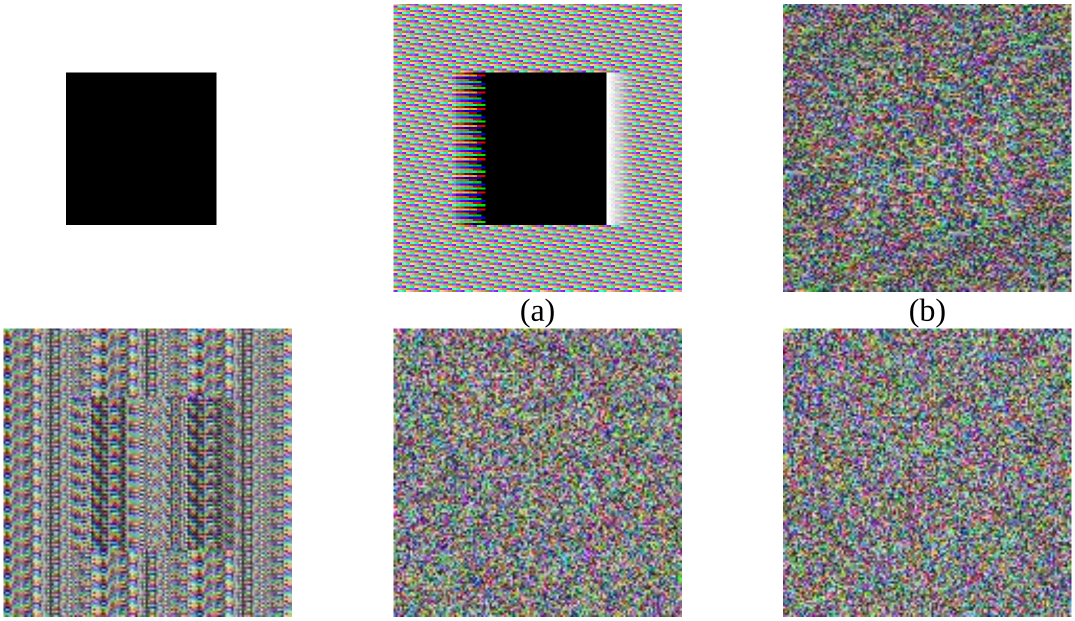


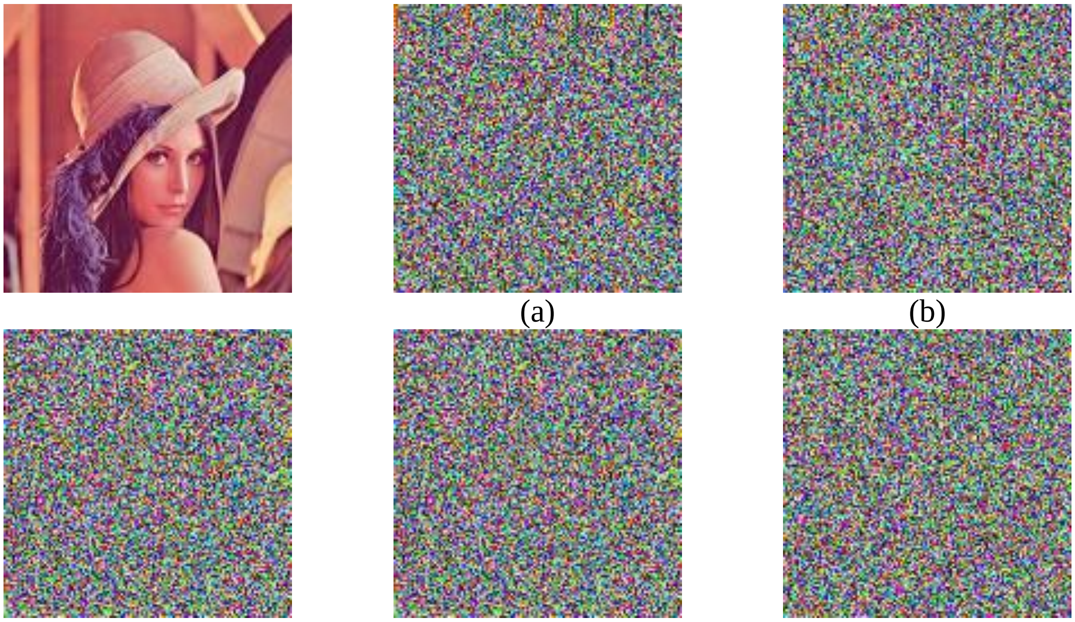
Figure 4.2: Symbol.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE.



(a) (b)

(c) (d) (e)

Figure 4.3: Blackbox.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE.



(a) (b)

(d) (d) (e)

Figure 4.4: Lena.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE.

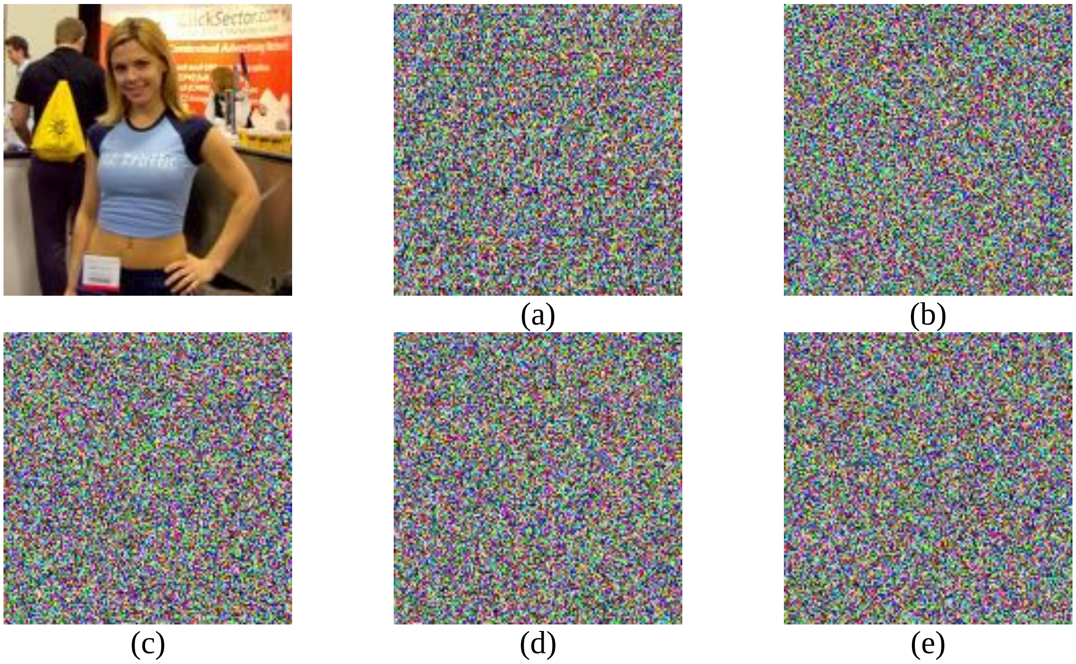


Figure 4.5: Girl.bmp encrypted by: a) HCM-PT, b) HCM-H, c) HCM-HMAC, d) HCM-EE, e) HCM-PRE.

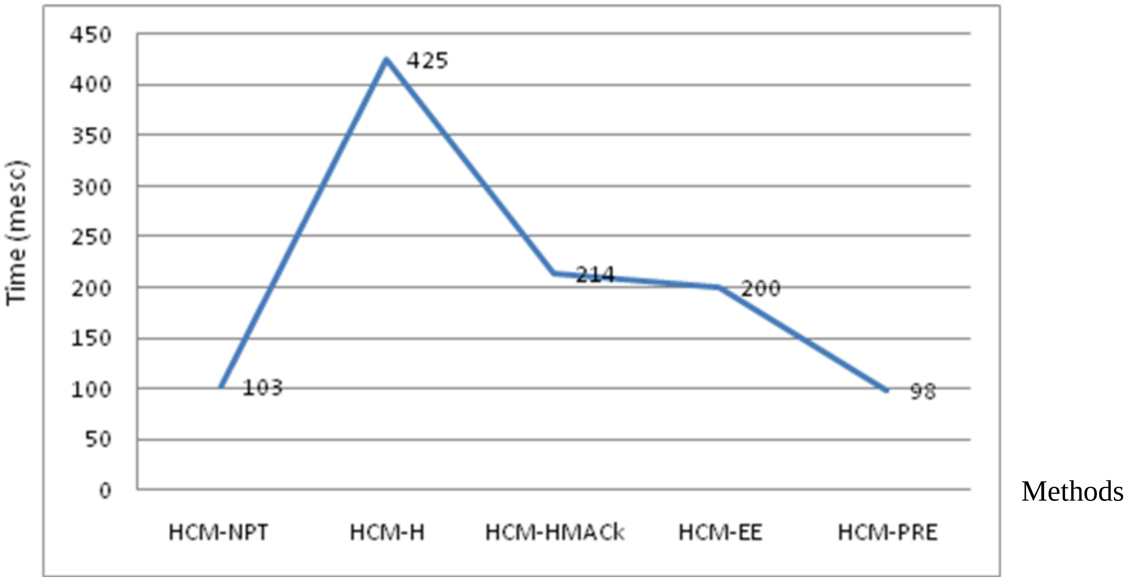


Figure 4.6: Encryption time (msec) of Nike.bmp with HCM-PT, HCM-H, HCM-HMAC, HCM-EE and HCM-PRE.

4.5. Security Analysis and Statistical Analysis

The ability to withstand all kinds of cryptanalysis and attacks [30]-[33] is a good measure of the performance of a cryptosystem. Robustness against attacks is used to evaluate the security of our schemes. It is shown that our proposed schemes are secure from the strongly cryptographic viewpoint. The results show the satisfactory security of the HCM-EE and HCM-PRE as explained and discussed in the following subsections.

4.5.1 Key Space Analysis

Key space is the total number of different keys that can be used in encryption. For a secure encryption scheme, the key space should be large enough to make brute force attacks infeasible. For the HCM-PRE, the key space is the same as that of HC [1][2]. Therefore the key space of the scheme is large; hence it is secure against brute force attack.

4.5.2 Known Plaintext-Ciphertext Attack KPCA

The KPCA is effective if the same key is used to encrypt many plaintexts. With the same reasoning of HCM-PT [16] and HCM-H[18], our proposed modifications HCM-PRE and HCM-EE are secure against the KPCA since each plaintext is encrypted by a different key, and the number of such dynamic keys is significantly large (4.11). Equations (3.5), (4.8), and (4.11) show that the NDK(HCM-PRE) (4.11) is larger than the NDK(HCM-PT) (3.5) and NDK(HCM-EE) (4.8); hence HCM-PRE is more secure.

4.5.3 Chosen-Ciphertext Attack

With the same reasoning of HCM-HMAC [20], the proposed HCM-EE and HCM-PRE are resistant against the chosen-ciphertext attack since all the shared parameters are exchanged via a secure protocol [6]. Knowledge of such parameters is necessary to accomplish the chosen-ciphertext attack. For the proposed HCM-EE and HCM-PRE, the opponent cannot use the exchanged parameters.

4.5.4 Statistical Analysis Resistance

In [34], it is mentioned that in [35] Shannon said “it is possible to solve many kinds of ciphers by statistical analysis”. A good cipher should be robust against any statistical attack. To prove the robustness of the proposed scheme, the statistical analysis has been performed. It is usually evaluated by the following measures [30][32][36][37][38]; calculating the histograms of the encrypted images and the correlation of two adjacent pixels in the plain/encrypted image demonstrating their superior confusion and diffusion property. The obtained results show that our scheme strongly withstands statistical attacks.

4.5.4.1 Histograms of encrypted images

We have calculated and analyzed the histograms of several encrypted images as well as their original images. Two typical examples are given in (Figs. 4.7-4.8). The histograms of the encrypted images are very close to uniform distribution; they are significantly different from those of the original image, and bear no statistical resemblance to the original image.

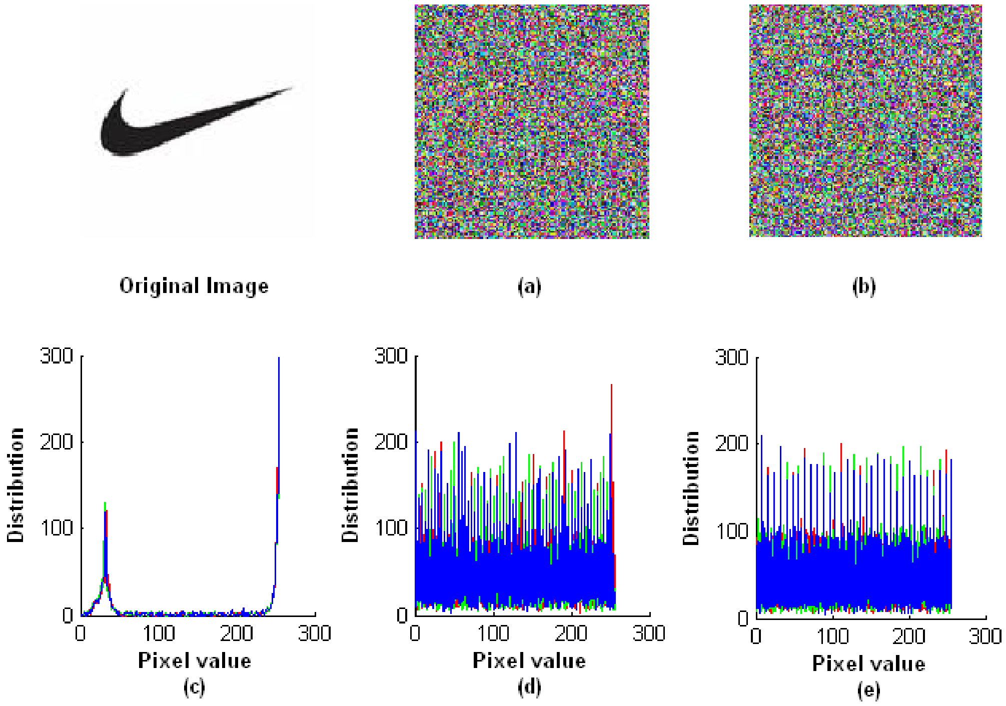


Figure 4.7: Histogram of RGB layers for original/encrypted Nike.bmp: a) HCM-EE-encrypted, b) HCM-PRE-encrypted, c) histogram of the original image, d) histogram of HCM-EE-encrypted e) histogram of HCM-PRE-encrypted

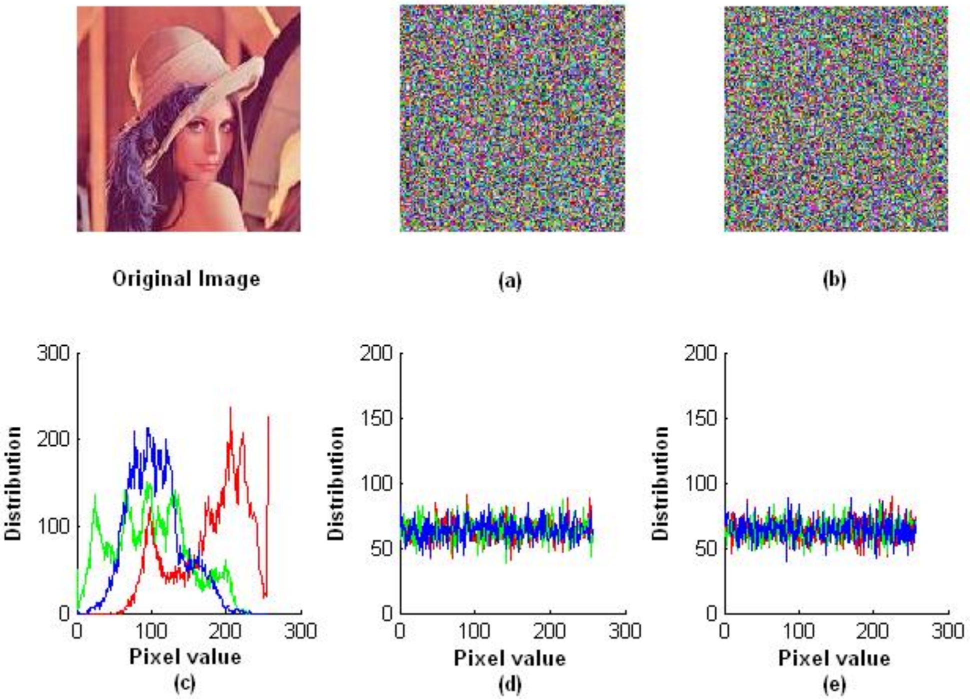


Figure 4.8: Histogram of RGB layers for original/encrypted Lena.bmp: a) HCM-EE-encrypted, b) HCM-PRE-encrypted, c) histogram of the original image, d) histogram of HCM-EE-encrypted e) histogram of HCM-PRE-encrypted

4.5.4.2 Correlation of Two Adjacent Pixels

There is a very good correlation between adjacent pixels in the plain-image (Nike.bmp: Figs. 4.1 and 4.9, Lena.bmp: Figs. 4.5 and 4.10). We studied the correlation between two adjacent pixels in plain-image and encrypted image in three different orientations (horizontal, vertical and diagonal). We use the following procedure: first 1000 pairs of two adjacent pixels in three different orientations are selected randomly from image to test correlation, and then using (2.1) calculate the correlation coefficient C.C of each pair. Figs. 4.9 and 4.10 show the correlation coefficients (explained in Chapter 2) of two adjacent pixels in Nike.bmp and Lena.bmp encrypted by HCM-EE, HCM-PRE, HCM-

NPT, HCM-H and HCM-HMAC in three different orientations as a practical example for different image types; Table 4.3 shows the numerical evaluation of the calculated correlations. It is clear that, the neighboring pixels in the plain-image have a very high correlation while they have a very small correlation (the closer to zero, the better) for encrypted images. This proves that the proposed encryption scheme HCM-PRE satisfies very small correlation and is better than other inspected schemes in the case of images with large single colour areas. We also note that the proposed scheme HCM-PRE versus HCM-EE gives alternately better correlation. On the other hand, the correlation values in Table 4.3 show that the examined schemes give nearly the same results in images containing many high frequency components.

Table 4.3: Correlation coefficients of two adjacent pixels in original and HCM-PT-encrypted images, HCM-H-encrypted images, HCM-HMAC-encrypted images, HCM-EE-encrypted images and HCM-PRE-encrypted images.

Image	Direction	Plain Image	Encrypted Image				
			HCM-PT	HCM-H	HCM-HMAC	HCM-EE	HCM-PRE
Nike.bmp	Horizontal	0.9413	0.0849	-0.0337	-0.0517	-0.0004	0.0287
	Vertical	0.9031	0.1484	0.0951	-0.0134	0.0396	0.0139
	Diagonal	0.9801	0.5743	-0.0602	0.0743	0.0365	0.0051
Lena.bmp	Horizontal	0.9349	-0.0074	0.0498	0.0168	0.0282	-0.0435
	Vertical	0.8538	0.0037	-0.0473	0.0026	0.0282	-0.0130
	Diagonal	0.8852	-0.0774	-0.0146	0.0071	0.0095	0.0084

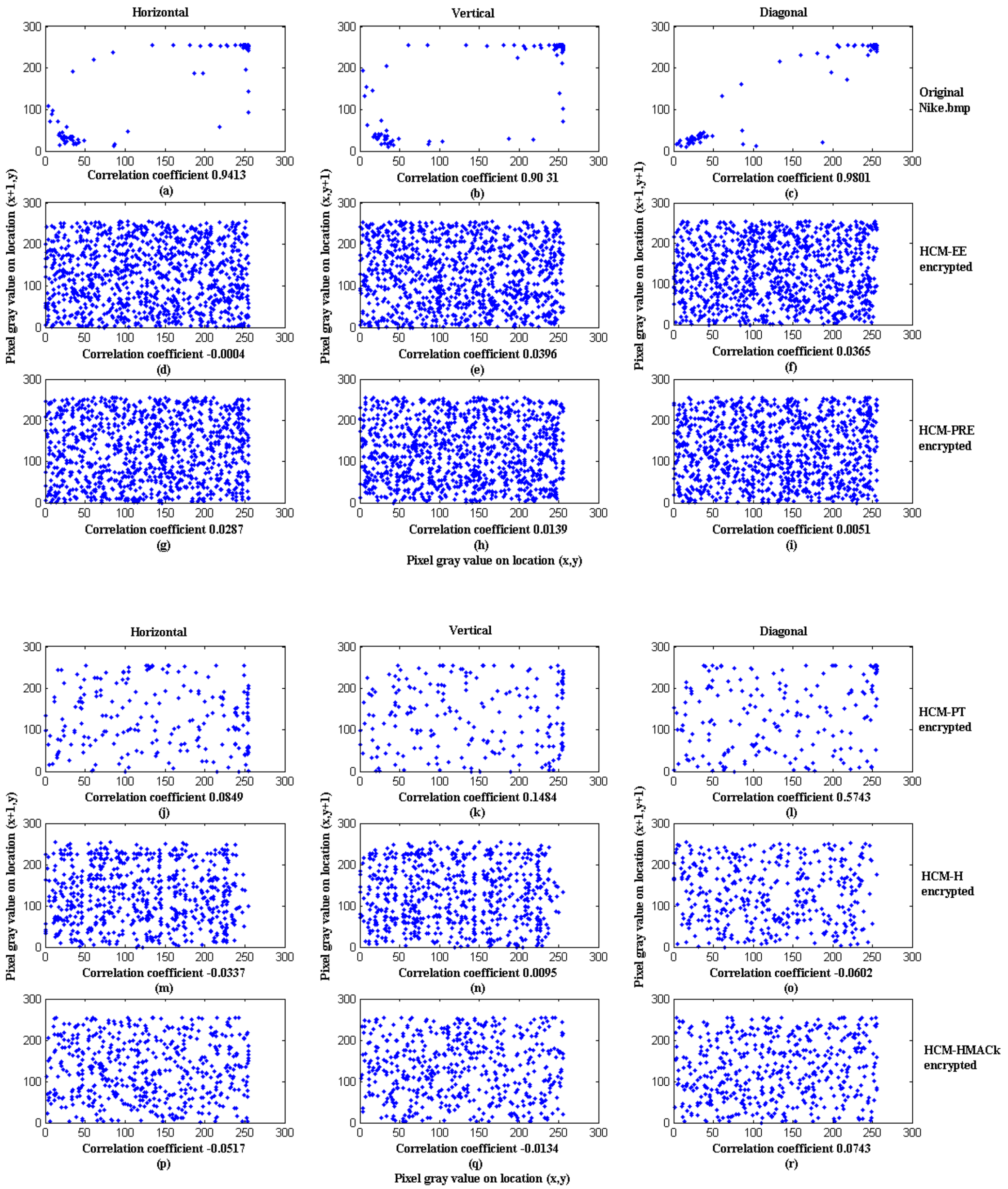


Figure 4.9: Correlation coefficients of two adjacent pixels in Nike.bmp encrypted by: HCM-EE, HCM-PRE, HCM-PT, HCM-H, and HCM-HMAC

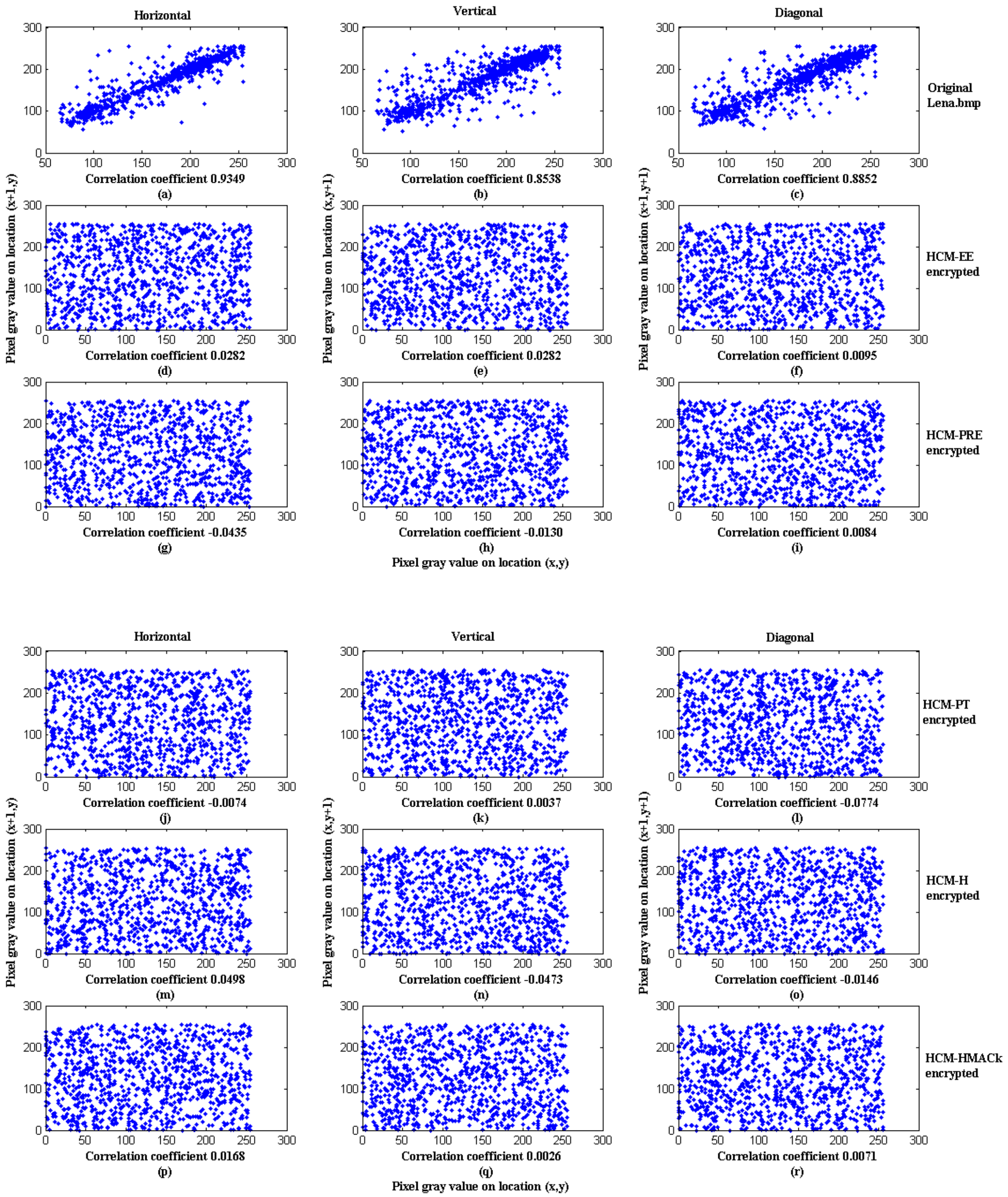
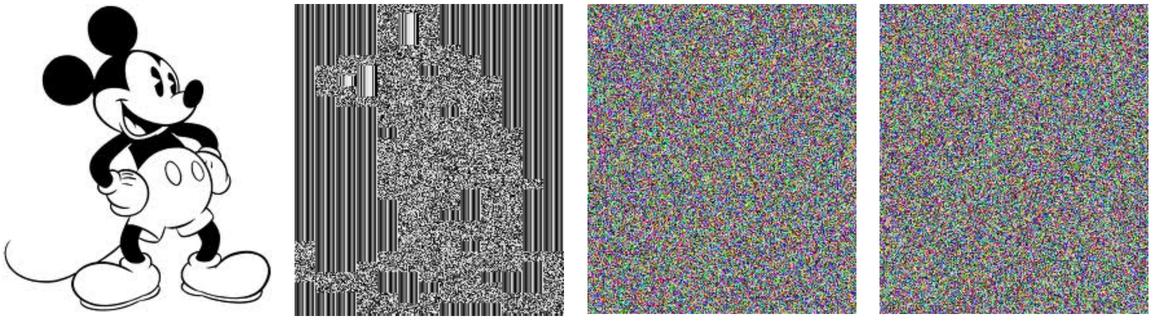


Figure 4.10: Correlation coefficients of two adjacent pixels in Lena.bmp encrypted by: HCM-EE, HCM-PRE, HCM-PT, HCM-H, and HCM-HMAC.

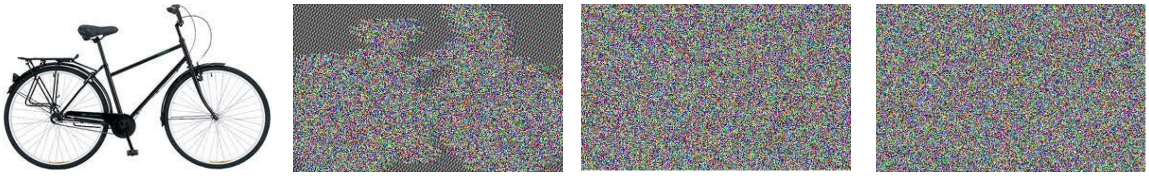
4.6 HCM-PRE and HCM-EE versus AES

To give adequate performance comparison, we examine our proposed HCM-PRE versus other well known algorithms (e.g. AES). We examined the encryption quality of several images. Based on visual inspection, the proposed HCM-PRE and HCM-EE encrypt the images with large single colour areas (identical plaintext blocks), they successfully hide data patterns. The AES fails to hide the data patterns for the images contain large single colour areas (Micky.bmp: Fig. 4.11, Bicycle.bmp: Fig. 4.12, and Penguin.bmp: Fig. 4.13). That is, the proposed HCM-PRE and HCM-EE have advantage in encryption of identical plaintext blocks over the AES.



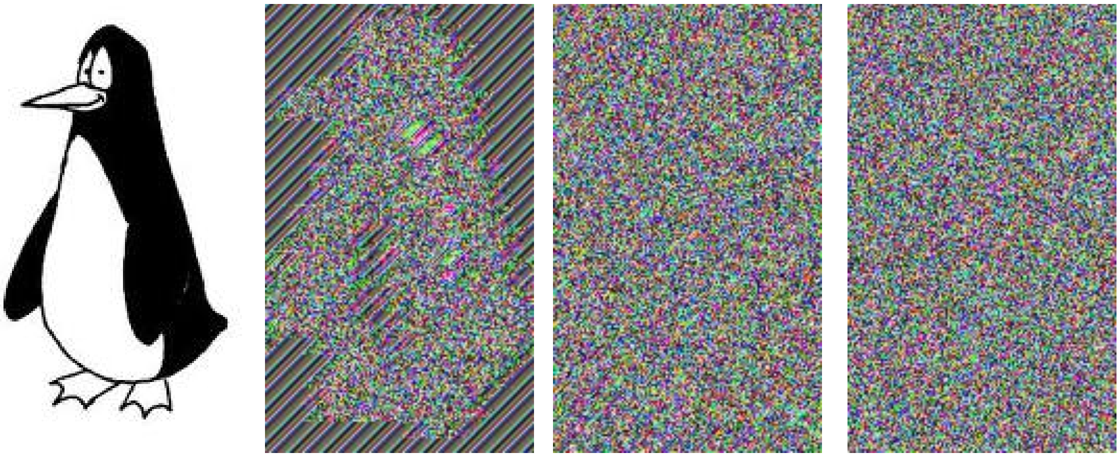
(a) (b) (c)

Figure 4.11: Mickey.bmp encrypted by: a) AES, b) HCM-EE, c) HCM-PRE.



(a) (b) (c)

Figure 4.12: Bicycle.bmp encrypted by: a) AES, b) HCM-EE, c)HCM-PRE.



(a) (b) (c)

Figure 4.13: Penguin.bmp encrypted by: a) AES, b) HCM-EE, c) HCM-PRE.

4.7 Conclusion

In this chapter, we have presented two new HC modifications, HCM-EE and HCM-PRE, based on the use of eigenvalues for generating a new key matrix for each plaintext block. In addition, five modifications of Hill cipher algorithms have been implemented for image encryption: HCM-PT, HCM-H, HCM-HMAC, and proposed here HCM-EE and HCM-PRE. Quality of image encryption for all algorithms is studied using visual inspection and numerical quality measures explained. From the obtained results, it is found that the proposed HCM-PRE is more effective in encryption quality than HCM-PT, HCM-H, HCM-HMAC, and HCM-EE. Encryption time for all the algorithms have been considered, the proposed HCM-PRE is about two times faster than HCM-EE and HCM-HMAC, and four times faster than the HCM-H.

The proposed modification HCM-PRE and HCM-EE resist the KPCA because of the use of dynamically changing key matrices similar to other considered here HC modifications (HCM-PT, HCM-NPT, HCM-H, HCM-HMAC) but the proposed HCM-PRE is more secure than HCM-H, HCM-PT, HCM-NPT, HCM-EE because of the significantly larger number of dynamic keys generated ((4.11) versus (4.8), (3.11) and (3.5)). Experimental analysis also shows that the HCM-EE and HCM-PRE resist the statistical attacks. The obtained results in this chapter thus encourage the use of the proposed schemes especially when image encryption is required.

Chapter 5

COMMUTATIVE MATRIX-BASED DIFFIE-HELLMAN-LIKE KEY EXCHANGE PROTOCOL

5.1 Introduction

Key exchange protocol has been intensively developed by several researchers. Public key cryptosystems do not need to have a shared secret between communicating parties. This solves the problem of large confidential communication network introduced earlier. Due to the increase in processor speed and even more due to smart modern cryptanalysis, the key size of public key cryptography grew very large. This created disadvantage in comparisons to symmetric key cryptosystems: public key cryptosystem is significantly slower, and requires large memory capacity and large computational power. As an example, 128-bit key used with DES [9] cryptosystem has approximately the same level of security as 1024-bit key used with RSA [11][29] public key cryptosystem. To solve these problems, researchers introduced different approaches. In order to decrease the key size so that public key cryptography can be used in smaller computational environments (such as smart cards or handheld wireless devices). The most common implementation solution is to combine symmetric key cryptosystems with public key cryptography. The idea, to overcome the problems related to applying the symmetric encryption only, the plaintext (multimedia component for example image) is encrypted using a fast symmetric key scheme, and only the secret key used for the symmetric encryption is encrypted with the slow public key scheme such as RSA.

The used key can be shared between the parties by the Diffie-Hellman (DH) Key exchange protocol [39].

In this chapter, we present our proposed a DH-like protocol using commutative matrices represented as conjugates to diagonal matrices [4]. The trap-door function exploited is a matrix multiplication with the zero-determinant matrix. In DH, a publicly known primitive element is utilized to obtain a public key from a private key. Similarly, in the proposed protocol, some publicly available zero-determinant matrix is used to construct a public key from the private ones. The commutativity of the matrices applied as private keys allows both parties to arrive at the same key by different ways of calculation. This is similar to the procedure in the DH protocol except that we multiply matrices instead of taking the exponentiation of big numbers.

The proposed protocol has high performance as computationally very simple because of applying few operations of multiplication to matrices whose entries are conventional numbers. Contrary to DH, even for 16×16 matrices with short 7-bit integer entries it provides substantial security of 2^{112} search space size.

The DH protocol was found to be susceptible to the intruder-in-the-middle attack [40] that led to invention of numerous DH extensions providing resistance to the attack. Currently, MQV protocol introduced by “Menezes, Qu and Vanstone” in [41] and HMQV (H for hash function) are considered as the most secure key-exchange protocols based on the DH [42][43]. Similarly, our DH-like protocol is also susceptible to the attack, and we show how to extend our protocol to a secure key-exchange protocol resembling MQV and HMQV.

In this thesis, we assume that the sender *A*, and the receiver *B* are using the proposed exchange protocols in [6][7] to exchange all the shared matrices and seed values required for HCM-PRE [3] and HCM-EE [4].

The original DH protocol is presented in section 5.2. Section 5.3 presents our DH-like key exchange protocol, and Section 5.4 analyses its security. Section 5.5 describes briefly MQV and HMQV and introduces a secure extension of our DH-like protocol that is similar to HMQV.

5.2 Overview of Key-Exchange Diffie-Hellman (DH)

The key-exchange Diffie-Hellman (DH) protocol without transfer of the secret key over an insecure channel was suggested by Diffie and Hellman in 1976 [39]. It is based on the complexity of discrete logarithm problem (DLP) solving over a finite field$GF(q)$, where q is prime. It utilizes a publicly available primitive element α of$GF(q)$.

Each user generates an independent secret random number X_i from a set of integers $\{1,2,...,q-1\}$but makes publicly available$Y_i = \alpha^{X_i} \bmod q$. If users i and j want to communicate privately, they use the value of$K_{ij} = \alpha^{X_i X_j} \bmod q = Y_i^{X_j} \bmod q = Y_j^{X_i} \bmod q$ as a secret key. This technique requires rather long numbers (200-bit big numbers are considered for estimation of its security as 2^{100} operations complexity in [39].

Conventional computers usually deal with 32- or 64-bit numbers. DH protocol was extended to matrix rings in [44], but it is still based on the complexity of the DLP. Its security was discussed in [45].

Further DH protocol matrix oriented modifications based on DLP are proposed and discussed [46][47][48][49][50]. A group-theoretic public-key exchange protocol is proposed [51]. This protocol [51] requires the transference of the full sets of group elements by both parties willing to get a common key and is based on the complexity of solving conjugacy equations. A non-commutative group-theoretical DH protocol extension using exponentiation is described in [52].

5.3 The DH-Like Protocol

Assume two communicating parties, $A_i, i = \overline{1,2}$, share two publicly available matrices

$$M \in GL(m,F), G \in GN(m,F), \tag{5.1}$$

where $GL(m,F)$ is the set of all invertible $m \times m$ matrices with entries from the field F with $|F|$ elements, and $GN(m,F)$ is the set of $m \times m$ matrices with entries from the field F and having rank $m-1$ and zero determinant value,

$$rank(G) = m-1, \det(G) = 0. \tag{5.2}$$

Assume that party A_i has two secret matrices (his private key)

$$X_{ij} = M^{-1} D_{ij} M \in GL(m,F), \tag{5.3}$$

where $D_{ij} \in GL(m,F)$ is a diagonal matrix, $i = \overline{1,2}, j = \overline{1,2}$. It is easy to see that these matrices commute:

$$\begin{aligned} X_{1i} X_{2j} &= M^{-1} D_{1i} M M^{-1} D_{2j} M \\ &= M^{-1} D_{1i} D_{2j} M \\ &= M^{-1} D_{2j} D_{1i} M \\ &= M^{-1} D_{2j} M M^{-1} D_{1i} M \\ &= X_{2j} X_{1i} \end{aligned}, \tag{5.4}$$

since diagonal matrices commute.

With the following protocol, the parties can obtain a key, K, shared by both parties without the transference of K.

5.3.1 The Protocol

1. User A_i calculates his public key

$$Y_i = X_{i1} G X_{i2}, \; i = \overline{1,2}. \tag{5.5}$$

2. User A_i sends his public key Y_i to the user A_{3-i}, $i = \overline{1,2}$.

3. User A_i calculates the common key $K = K_1 = K_2$ using his private key and the received public key of his partner

$$K_i = X_{i1} Y_{3-i} X_{i2}, \; i = \overline{1,2}. \tag{5.6}$$

The protocol results in the same value $K = K_1 = K_2$ for both parties since, due to (5.4)-(5.6), the following holds:

$$K_1 = X_{11} Y_2 X_{12} = X_{11} X_{21} G X_{22} X_{12} = X_{21} X_{11} G X_{12} X_{22} = X_{21} Y_1 X_{22} = K_2 .$$

Note that due to (5.2) and (5.3)

$$rank(K) = m - 1, \det(K) = 0 . \tag{5.7}$$

<u>Example:</u> Let $m = 2, F = Z_5 = \{0,1,..,4\}$, and according to (5.2), (5.3),

$$M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \; M^{-1} = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}, \; G = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix},$$

$$D_{11} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}, D_{12} = \begin{bmatrix} d_3 & 0 \\ 0 & d_4 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}, \; D_{21} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \; D_{22} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}. \tag{5.8}$$

Then from (5.3) and (5.8) one calculates

$$X_{11} = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 4 & 8 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix},$$

$$X_{12} = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 4 \\ 8 & 8 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 2 & 3 \end{bmatrix}, \tag{5.9}$$

$$X_{21} = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 1 & 4 \end{bmatrix},$$

$$X_{22}=\begin{bmatrix}3&1\\4&2\end{bmatrix}\begin{bmatrix}3&0\\0&2\end{bmatrix}\begin{bmatrix}1&2\\3&4\end{bmatrix}=\begin{bmatrix}4&2\\2&4\end{bmatrix}\begin{bmatrix}1&2\\3&4\end{bmatrix}=\begin{bmatrix}0&1\\4&0\end{bmatrix}.$$

And according to (5.5), from (5.9)

$$\begin{aligned}Y_1&=\begin{bmatrix}0&2\\3&0\end{bmatrix}\begin{bmatrix}1&3\\4&2\end{bmatrix}\begin{bmatrix}3&3\\2&3\end{bmatrix}=\begin{bmatrix}3&4\\3&4\end{bmatrix}\begin{bmatrix}3&3\\2&3\end{bmatrix}=\begin{bmatrix}2&1\\2&1\end{bmatrix},\\Y_2&=\begin{bmatrix}4&4\\1&4\end{bmatrix}\begin{bmatrix}1&3\\4&2\end{bmatrix}\begin{bmatrix}0&1\\4&0\end{bmatrix}=\begin{bmatrix}0&0\\2&1\end{bmatrix}\begin{bmatrix}0&1\\4&0\end{bmatrix}=\begin{bmatrix}0&0\\4&2\end{bmatrix}.\end{aligned}\tag{5.10}$$

Finally, by (5.6) and (5.10),

$$\begin{aligned}K_1&=\begin{bmatrix}0&2\\3&0\end{bmatrix}\begin{bmatrix}0&0\\4&2\end{bmatrix}\begin{bmatrix}3&3\\2&3\end{bmatrix}=\begin{bmatrix}3&4\\0&0\end{bmatrix}\begin{bmatrix}3&3\\2&3\end{bmatrix}=\begin{bmatrix}2&1\\0&0\end{bmatrix},\\K_2&=\begin{bmatrix}4&4\\1&4\end{bmatrix}\begin{bmatrix}2&1\\2&1\end{bmatrix}\begin{bmatrix}0&1\\4&0\end{bmatrix}=\begin{bmatrix}1&3\\0&0\end{bmatrix}\begin{bmatrix}0&1\\4&0\end{bmatrix}=\begin{bmatrix}2&1\\0&0\end{bmatrix}.\end{aligned}\tag{5.11}$$

Thus, from (5.11) one gets $K=K_1=K_2$, that meets (5.7).

5.4 Security Analysis of the Protocol

An opponent knowing M ,G and viewing $Y_i, i=\overline{1,2}$, is not able to obtain K because he needs the secret matrices $X_{ij}, i=\overline{1,2}, j=\overline{1,2}$ for that purpose. He can try to get them by substituting (5.3) into (5.5) and trying to solve the resulting system of nonlinear algebraic equations with respect to the $2m$ unknown diagonal elements $D_{i1}(l,l), D_{i2}(l,l), l=\overline{1,m}$. For the example above, (5.3) and (5.8) yield the following private key matrices

$$\begin{aligned}X_{11}&=\begin{bmatrix}3&1\\4&2\end{bmatrix}\begin{bmatrix}d_1&0\\0&d_2\end{bmatrix}\begin{bmatrix}1&2\\3&4\end{bmatrix}=\begin{bmatrix}3d_1&d_2\\4d_1&2d_2\end{bmatrix}\begin{bmatrix}1&2\\3&4\end{bmatrix}=\begin{bmatrix}3(d_1+d_2)&d_1+4d_2\\4d_1+d_2&3(d_1+d_2)\end{bmatrix},\\X_{12}&=\begin{bmatrix}3&1\\4&2\end{bmatrix}\begin{bmatrix}d_3&0\\0&d_4\end{bmatrix}\begin{bmatrix}1&2\\3&4\end{bmatrix}=\begin{bmatrix}3d_3&d_4\\4d_3&2d_4\end{bmatrix}\begin{bmatrix}1&2\\3&4\end{bmatrix}=\begin{bmatrix}3(d_3+d_4)&d_3+4d_4\\4d_3+d_4&3(d_3+d_4)\end{bmatrix},\end{aligned}$$

and the following matrix equation

$$Y_1 = \begin{bmatrix} 3(d_1+d_2) & d_1+4d_2 \\ 4d_1+d_2 & 3(d_1+d_2) \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 3(d_3+d_4) & d_3+4d_4 \\ 4d_3+d_4 & 3(d_3+d_4) \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix},$$

or

$$\begin{aligned} Y_1 &= \begin{bmatrix} 2d_1+4d_2 & d_1+2d_2 \\ d_1+3d_2 & 3d_1+4d_2 \end{bmatrix} \begin{bmatrix} 3(d_3+d_4) & d_3+4d_4 \\ 4d_3+d_4 & 3(d_3+d_4) \end{bmatrix} \\ &= \begin{bmatrix} 2d_1d_4+4d_2d_4 & d_1d_4+2d_2d_4 \\ d_1d_4+3d_2d_4 & 3d_1d_4+4d_2d_4 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix} \end{aligned} \qquad (5.12)$$

is obtained.

Denoting

$$z_1 = d_1d_3, z_2 = d_1d_4, z_3 = d_2d_3, z_4 = d_2d_4, \qquad (5.13)$$

the following system of equations can be derived from (5.12)

$$\begin{aligned} &2z_2 + 4z_4 = 2, \\ &z_2 + 2z_4 = 1, \\ &z_2 + 3z_4 = 2, \\ &3z_2 + 4z_4 = 1. \end{aligned} \qquad (5.14)$$

From the last two equations of (5.14) one can write

$$z_2 = 2 + 2z_4. \qquad (5.15)$$

The first two equations of (5.14) give

$$z_2 = 1 + 3z_4. \qquad (5.16)$$

Hence, (5.15) and (5.16) imply

$$z_4 = 1, z_2 = 4\,. \qquad (5.17)$$

The other unknowns, z_1, z_3, are not defined. From (5.13) and (5.17), it follows that

$$z_4 = 1 = d_2d_4, z_2 = 4 = d_1d_4\,. \qquad (5.18)$$

From (5.18), one gets

$$4d_2 = d_1\,. \qquad (5.19)$$

It is easy to check that D_{11} defined in (5.8) meets (5.19).

As in the example above, system (5.5) after taking (5.3) into account may be solved as a system of m^2 linear algebraic equations by introducing m^2 unknowns $z_{ilk} = D_{i1}(l,l) \cdot D_{i2}(k,k)$, $l = \overline{1,m}, k = \overline{1,m}$. However, due to (5.2), one of its rows is a linear combination of its other rows. Hence, the right-hand side of (5.5) has one row that is a linear combination of its other rows. The system of m^2 equations therefore has $m^2 - m$ linear independent equations. It is thus undetermined and its solution has m free unknowns, enumeration of whose possible values is required to find a solution of (5.3), (5.5). In the worst case this enumeration requires checking of $|F|^m$ variants. If e.g., $|F| = 128, m = 16$, then the number of variants to check is 2^{112} which is unfeasible for the current level of computer development.

5.5 Matrix-based DH-like Secure Protocol Extension

The proposed matrix-based DH-like protocol bares the same deficiencies as the original DH key-exchange protocol does, e.g., it is susceptible to the intruder-in-the-middle attack [40]. Secure protocols extending DH original protocol are considered, e.g., in [40][43]. These algorithms use some information known in advance to the both communicating parties (passwords, static keys). These secure protocols are based on the original DH key-exchange protocol and may be used in multiplicative (for the Discrete Logarithm Problem) or in additive (Elliptic Curves) fields [43, p.3].

Secure MQV and HMQV protocols are described in [43, Figs. 5.1, 5.2]. Assuming that certified static public keys, $A = g^a, B = g^b$, (calculated using respective private keys $a,b \in Z_q$) of the communicating parties, $\hat{A}, \hat{B}$, respectively, are known to the parties in advance, MQV and HMQV may be represented by Fig. 5.1.

$$
\begin{aligned}
&\hat{A} : generate\ x, X = g^{x};\\
&\hat{B} : generate\ y, Y = g^{y};\\
&\hat{A} \rightarrow \hat{B} : \hat{A}, \hat{B}, X\\
&\hat{B} \rightarrow \hat{A} : \hat{B}, \hat{A}, Y\\
&\hat{A} : \sigma_{\hat{A}} = (YB^{e})^{x+da}; K = H(\sigma_{\hat{A}})\\
&\hat{B} : \sigma_{\hat{B}} = (XA^{d})^{y+eb}; K = H(\sigma_{\hat{B}})\\
&MQV : d = 2^{l} + X \bmod 2^{l}, e = 2^{l} + Y \bmod 2^{l};\\
&l = \lceil \log_2 q \rceil / 2\\
&HMQV : e = H(X \,\|\, \hat{B}), d = H(Y \,\|\, \hat{A})
\end{aligned}
$$

Figure 5.1: Computation of the session key K by MQV and HMQV (H is a hash function).

Contrary to the original DH key-exchange protocol using some generated by users secret numbers (ephemeral, dynamic keys) x, y, and respective public values X, Y, only, their static private, a, b, and public, A, B, keys are used in MQV and HMQV (Fig. 5.1) thus not allowing an opponent to apply intruder-in-the-middle attack.

Matrix analogue of HMQV may be represented by Fig. 5.2 assuming that public keys, $A = A_1GA_2, B = B_1GB_2$ (generated from respective private keys, $a = (A_1, A_2), b = (B_1, B_2)$) are known to the both communicating parties in advance. They generate their ephemeral secret keys, calculate respective public keys, exchange them, and use both static and ephemeral keys to calculate common session key K (Fig. 5.2).

$$
\begin{aligned}
&\hat{A} : generate\ x = (X_1, X_2), X = X_1 G X_2; \\
&\hat{B} : generate\ y = (Y_1, Y_2), Y = Y_1 G Y_2; \\
&\hat{A} \rightarrow \hat{B} : \hat{A}, \hat{B}, X \\
&\hat{B} \rightarrow \hat{A} : \hat{B}, \hat{A}, Y \\
&\hat{A} : \sigma_{\hat{A}} = eX_1 B X_2 + dA_1 Y A_2; K = H(\sigma_{\hat{A}}); \\
&\hat{B} : \sigma_{\hat{B}} = eB_1 X B_2 + dY_1 A Y_2; K = H(\sigma_{\hat{B}}) \\
&e = H(X \parallel \hat{B}), d = H(Y \parallel \hat{A})
\end{aligned}
$$

Figure 5.2: Matrix-based analogue of HMQV

Hence, our protocol opens a door for a variety of similar secure protocols which are however less computationally intensive than DLP- or Elliptic Curve-based protocols since they apply few simple matrix multiplications only.

5.6 Conclusion

The proposed DH-like matrix protocol is based on few matrix multiplications and does not use exponentiation as do other known DH protocol modifications. The concept of the proposed protocol is the same as that of DH: it allows two-way arrival at the same common key that is provided by the use of private key matrices commuting each other as conjugates to diagonal invertible matrices. The public key is obtained by multiplication of the private key matrices with a publicly known zero-determinant matrix. The non-invertibility of this matrix defines the trap-door property of our protocol. For 16×16 matrices with 7-bit integer entries it ensures substantial security of 2^{112} search space size. The proposed protocol, contrary to previous ones, may operate with short numbers and is computationally simple thus assuring its high performance and wide applicability. The proposed matrix-based DH-like protocol bares the same deficiencies as the original DH key-exchange protocol, e.g., it is susceptible to the intruder-in-the-middle attack [40]. Secure protocols extending DH original protocol are considered, e.g., [40][43]. These algorithms (e.g., MQV, HMQV) utilize some information known in advance to both communicating parties (passwords, static keys) and may be exploited in multiplicative fields (for the Discrete Logarithm Problem), or in additive fields (Elliptic Curves) [43 p.3]. Herein, we propose a protocol similar to HMQV that is based on our DH-like protocol. Hence, our protocol opens a door for a variety of similar secure protocols which are however less computationally intensive than DLP- or Elliptic Curve-based protocols since they use few simple matrix multiplications.

Chapter 6

ELGAMAL PUBLIC KEY CRYPTOSYSTEM AND SIGNATURE SCHEME IN $GU(m,p,n)$

6.1 Introduction

In this chapter, we present our proposed ElGamal public key cryptosystem and signature scheme $GU(m,p,n)$ [7]. It uses the group $GU(m,p,n)$ of numbers co-prime to mp^n and having analytical representation and known order. Elements of $GU(m,p,n)$ with the maximal order are used as the base elements in the proposed extension instead of primitive roots used in the original scheme. Proposed scheme allows easy periodic change of the group and base elements to provide necessary security level without change of the prime number p contrary to the case of $GF(p)$ used in the original ElGamal scheme. Computation of discrete logarithms in the proposed scheme is difficult for large p .

The rest of the chapter is organized as follows. Section 2 introduces an overview of ElGamal public key cryptosystem and signature. Section 3 introduces $GU(m,p,n)$ and its properties used in [28]. Section 4 presents the extension of ElGamal public key cryptosystem and signature scheme to$GU(m,p,n)$. Section 5 concludes the chapter.

6.2 Overview of Original ElGamal Public Key Cryptosystem and Signature Scheme

ElGamal public key cryptosystem and signature scheme [53] rely on the computational complexity of finding discrete logarithms base some publicly known primitive root (base element), $\alpha \in GF(p)$, where p is a large prime such that the maximal in $GF(p)$ element order, $mo_{GF(p)} = p-1$, has at least one large prime factor (if $mo_{GF(p)}$ has only small prime factors, discrete logarithm computation is easy [54]). A three-parametric finite number group $GU(m,p,n)$, a subgroup of the group $U(mp^n)$ of units [55] consisting of numbers co-prime to mp^n where p is prime and $m \neq 0 \bmod p$ introduced in [28]. The group $GU(m,p,n)$ (contrary to $GF(p)$ where finding primitive roots is computationally difficult [56]) has p^{n-1} a priori analytically known numbers of the maximal order $mo_{GU(m,p,n)} = p^{n-1}$, any of which can be used as a base element $\alpha \in GU(m,p,n)$, and if p is large, finding discrete logarithms [54] is difficult. We propose an extension of ElGamal public key cryptosystem and digital signature to $GU(m,p,n)$ that is beneficial because, due to its known analytical representation, the field and base element can be easily periodically changed as recommended in [57] not changing p (finding of new primes is difficult [58]), and message space can be made arbitrarily large also by respective choice of the parameters (messages are from Z_{mp^n})

6.3 Group $GU(m,p,n)$ and its Properties

Theorem 1:

If $p > 1$ is prime, and $m \bmod p \neq 0$, $n \geq 2$ are integers, then the set of numbers

$$GU(m,p,n) = \{1, mp^n - 1\} \cup \{qmp^{n-k} + b \neq 1 \mid q \neq 0 \bmod p,$$
$$|b| = 1,\ 1 \leq q < p^k, 1 \leq k < n\}$$

is a group with respect to multiplication modulo mp^n. The order of the group is

$$|GU(m,p,n)| = \begin{cases} 2p^{n-1}, (m > 1) \vee (p > 2) \\ 2^{n-1}, (m = 1) \& (p = 2) \end{cases}.$$

Example 1:

a) If $m = 2, p = 3,\ n = 5, k_1 = 2, k_2 = 3,\ q_1 = 2, b_1 = -1, q_2 = 4,\ \ b_2 = 1$, then $e_1 = q_1 mp^{n-k_1} + b_1 = 2 \cdot 2 \cdot 3^{5-2} - 1 = 107,\ \ e_2 = q_2 mp^{n-k_2} + b_2 = 4 \cdot 2 \cdot 3^{5-3} + 1 = 73$, and

b) $|GU(1,2,4)| = |\{1,3,5,7,9,11,13,15\}| = 2^3 = 8.$

c) $|GU(2,3,3)| = |\{1,5,7,11,13,17,19,23,25,29,31,35,37,41,43,47,49,53\}| = 2 \cdot 3^2 = 18.$

d) $|GU(3,2,3)| = |\{1,5,7,11,13,17,19,23\}| = 2 \cdot 2^2 = 8.$

e) $|GU(1,3,2)| = |\{1,2,4,5,7,8\}| = 2 \cdot 3 = 6.$

f) $|GU(1,2,2)| = |\{1,3\}| = 2^1 = 2.$

Theorem 2:

In the conditions of Theorem 1, the order of $qmp^{n-k} + b \in GU(m,p,n) - \{1, mp^n - 1\}$ is as follows:

$$ord(qmp^{n-k} + b) = \begin{cases} 2^{n-2}, (p = 2) \& (k = n-1) \& (n > 2), \\ 2p^k, (p > 2) \& (b = -1) \& (1 \leq k < n) \& (n \geq 2), \\ p^k, (p > 2) \& (b = 1) \& (1 \leq k < n) \& (n \geq 2) \vee (p = 2) \& (1 \leq k < n-1) \& \\ \qquad (n > 2), \end{cases}$$

and $ord(1) = 1, ord(mp^n - 1) = 2$.

Example 2:

Consider $GU(2,3,10)$. If $q = 1, k = n - 1 = 9, b = 1$, then $qmp^{n-k} + 1 = 7$, and its 3^9-th power (3^9=19683) modulo $2 \cdot 3^{10}$ must be equal to 1. Actually, 7^{19683}mod $2 \cdot 3^{10}$ = 1,16070431941779186268940237099835e+16634 mod 118098 =1, and 7^{19682} mod

$2\cdot 3^{10}$ = 1,6581490277397026609848605299976e+16633 mod 118098 =101227.

Let $GU_L(m,p,n)=\{e\in GU(m,p,n)-\{1,mp^n-1\}\,|\,ord(e)=L\}$, then in the conditions of **Theorem 1**,

$$|GU_L(m,p,n)|=\begin{cases} p^{k-1}(p-1),((L=p^k)\vee(L=2p^k))\,\&\,(1\le k<n)\,\&\,(p>2),\\ 2^k,(L=2^k)\,\&\,(p=2)\,\&\,(1\le k<n-2)\,\&\,(m\ge 1),\\ 2^{n-1}+2^{n-2},(L=2^{n-2})\,\&\,(p=2)\,\&\,(m>1),\\ 2^{n-2},(L=2^{n-2})\,\&\,(p=2)\,\&\,(m=1).\end{cases}$$

Example 3:

a) Consider $GU(3,2,3)=\{1,5,7,11,13,17,19,23\}$. There are $6=2^2+2$ numbers of order $2=2^{3-2}$: $|GU_2(3,2,3)|=|\{5,7,11,13,17,19\}|=6$.

b) Consider $GU(1,2,4)=\{1,3,5,7,9,11,13,15\}$. There are $4=2^{4-2}$ numbers of order $4=2^{4-2}$: $|GU_4(1,2,4)|=|\{3,5,11,13\}|=4$, and $2=2^1$ numbers of order $2=2^1$: $|GU_2(1,2,4)|=|\{7,9\}|=2$ (not taking into account 15).

Consider $GU(2,3,3)=\{1,5,7,11,13,17,19,23,25,29,31,35,\ 37,41,43,47,49,53\}$. There are $6=3^{2-1}(3-2)$ numbers of order $9=3^2$: $|GU_{3^2}(2,3,3)|=|\{7,13,25,31,43,49\}|$ $=3^{2-1}(3-1)=6$, the same number of numbers of order $18=2\cdot 3^2$: $|GU_{2\cdot 3^2}(2,3,3)|=|\{5,11,23,29,41,47\}|=3^{2-1}(3-1)=6$,

$2=3^{1-1}(3-1)$ numbers of order $3=3^1$: $|GU_3(2,3,3)|=|\{19,37\}|=3^{1-1}(3-1)=2$, and the same number of numbers of order $6=2\cdot 3^1$: $|GU_6(2,3,3)|=|\{17,35\}|=3^{1-1}(3-1)=2$.

6.4 ElGamal Public Key Cryptosystem and Signature Scheme in $GU(m,p,n)$

6.4.1 Public key cryptosystem

Suppose that A wants to send B a message M, where $M \in Z_{mp^n}$. First, A chooses a number l uniformly from $Z_{mp^n} - \{0,1\} = \{2,..,mp^n - 1\}$. Then A computes the key

$$K = y_B^l \bmod mp^n, \tag{6.1}$$

where $y_B = \alpha^{X_B} \bmod mp^n$, is a secret value of *B*,

$$\alpha = qmp + 1 \in GU(m,p,n), q \neq 0 \bmod p, 1 \leq q < p^{n-1}, \tag{6.2}$$

$ord(\alpha) = mo_{GU(m,p,n)} = p^{n-1}$, according to Theorem 2, $X_B \in Z_{mp^n} - \{0,1\}$, and y_B is kept in the public file of B together with α. The ciphertext is then the pair (c_1, c_2), where

$$c_1 = \alpha^l \bmod mp^n, c_2 = KM \bmod mp^n. \tag{6.3}$$

Decryption splits into two parts. The first step is recovering *K*, which is easy for *B*, since $K = (\alpha^l)^{X_B} \bmod mp^n = c_1^{X_B} \bmod mp^n$, and X_B is known to *B* only. The second step is to multiply c_2 by $K^{-1} \bmod mp^n$, and recover the message M.

Example 4:

Consider $GU(2,3,10)$. Let $q = 2, k = n - 1 = 9, b = 1$, and $M = 3 \in Z_{mp^n} = Z_{118098}$, $l = 3$, $\alpha = 2 \cdot 2 \cdot 3 + 1 = 13 \in GU(2,3,10)$, $X_B = 6$,

$y_B = \alpha^{X_B} = 13^6 \bmod 2 \cdot 3^{10} = 4826809 \bmod 118098 = 102889$,

$K = y_B^l \bmod mp^n = 102889^3 \bmod 118098 = 9289$,

$c_1 = \alpha^l \bmod mp^n = 13^3 \bmod 118098 = 2197, c_2 = KM \bmod mp^n$ $= 9289 \cdot 3 \bmod 118098 = 27867$. Side *B* restores the message (*c₁, c₂*) by the following calculations: $K = c_1^{X_B} \bmod mp^n = (2197)^6 \bmod 118098 = 9289$,

$K^{-1} = 6751 \bmod 118098$, $M = K^{-1} \cdot c_2 \bmod mp^n = (6751 \cdot 27867) \bmod 118098 = 3$.

6.4.2 Signature Scheme

Let $M \in Z_{mp^n}$ is a document to be signed. The public file still consists of α and

$$y_a = \alpha^{X_a} \bmod mp^n, \tag{6.4}$$

for each user *a*, $X_a \in Z_{mp^n} - \{0,1\}$. The signature for M is the pair *(r,s)*, $r \in GU(m,p,n)$, $s \in Z_{mp^n}$, chosen such that the equation

$$\alpha^M = y_a^r r^s \bmod mp^n \tag{6.5}$$

is satisfied, where

$$r = \alpha^k \bmod mp^n,$$

$k \in U(mp^n)$ is a random number selected secretly, $U(N)$ is a group of numbers co-prime to N, and *s* is obtained as a solution of (6.5) written as

$$\alpha^M = \alpha^{X_a r} \cdot \alpha^{ks} \bmod mp^n,$$

from which it follows that

$$s = (M - X_a r)k^{-1} \bmod mp^n. \tag{6.6}$$

Given M, *r*, and *s*, it is easy to verify the authenticity of the signature by checking (6.5) using (6.4). The use of inverse modulo mp^n in (6.6) is possible since, by (6.2) and Theorem 2, $\alpha^{p^{n-1}} = 1 \bmod mp^n$ and p^{n-1} divides mp^n.

Example 5:

a) Let $m = 2, p = 3, n = 10$, $mp^n = 118098$, $M = 2 \in Z_{mp^n} = Z_{2 \cdot 3^{10}} = Z_{118098}$,

$k = 7 \in U(mp^n) = U(118098), X_a = 5 \in Z_{mp^n} - \{0,1\}, \alpha = 2 \cdot 2 \cdot 3 + 1 = 13 \in GU(2,3,10)$,

$$y_a = \alpha^{X_a} \bmod mp^n = 13^5 \bmod 118098 = 16999, r = 13^7 \bmod 118098 = 38479$$
$$k^{-1} = 101227 \bmod 118098, s = (M - X_a r)k^{-1} \bmod 2 \cdot 3^{10} = (2 - 5 \cdot 38479) \cdot 101227 \bmod 118098 = -61297 \bmod 118098 = 56871,$$

$$\alpha^M \bmod mp^n = 13^2 \bmod 118098 = 169 , \tag{6.7}$$

$$\begin{aligned} y_a^r \bmod mp^n &= 16999^{38479} \bmod 118098 \\ &= (16999^{10000} \bmod 118098)^3 \cdot 16999^{8479} \bmod 118098) \bmod 118098 , \\ &= 112555 \cdot 86335 \bmod 118098 = 96289 \end{aligned}$$

$$\begin{aligned} r^s \bmod mp^n &= 38479^{56871} \bmod 118098 \\ &= (38479^{5000} \bmod 118098)^{11} \cdot (38479^{1871} \bmod 118098) , \\ &= 35077 \cdot 16369 \bmod 118098 = 101035 \end{aligned}$$

$$y_a^r \cdot r^s \bmod mp^n = 96289 \cdot 101035 \bmod 118098 = 169 . \tag{6.8}$$

From (6.7), (6.8), it follows that (6.5) holds and the message $M = 2$ is correct.

b) Let $m = 2, p = 5, n = 7$, $mp^n = 156250$, $M = 2 \in Z_{mp^n} = Z_{156250}$,

$k = 3 \in U(mp^n) = U(2 \cdot 5^7) = U(156250)$, $X_a = 5 \in Z_{mp^n} - \{0,1\} = Z_{156250} - \{0,1\}$,

$\alpha = 2 \cdot 2 \cdot 5 + 1 = 21 \in GU(2,5,7)$, $y_a = \alpha^{X_a} \bmod mp^n = 21^5 \bmod 156250 = 21601$,

$r = \alpha^k \bmod mp^n = 21^3 \bmod 156250 = 9261$, $k^{-1} = 104167 \bmod 156250$,

$$\begin{aligned} s &= (M - X_a r)k^{-1} \bmod 5^6 = (2 - 5 \cdot 9261) \cdot 104167 \bmod 156250 \\ &= -119601 \bmod 15625 = 36649, \end{aligned}$$
$$y_a^r \bmod mp^n = 21601^{9261} \bmod 156250 = 56351,$$

$$r^s \bmod mp^n = 9261^{36649} \bmod 156250 = ((9261^{10000} \bmod 156250)^3 \cdot$$
$$(9261^{6649} \bmod 156250)) \bmod 156250 = 50001 \cdot 103841 \bmod 156250$$
$$= 122591,$$
$$y_a^r \cdot r^s \bmod mp^n = 56351 \cdot 122591 \bmod 156250 = 441. \tag{6.9}$$

6.5 Conclusion

The ElGamal public key cryptosystem and signature scheme originally proposed for $GF(p)$ are redefined for the group $GU(m,p,n)$. The elements of $GF(p)$ having the maximal order $p-1$(primitive roots) are the base elements of the original ElGamal system. Finding primitive roots in $GF(p)$ is computationally difficult [56]. For the group $GU(m,p,n)$, p^{n-1} a priori analytically known its elements represented by (6.2) have the maximal order of p^{n-1}. They are used as the base elements in the proposed system that allows easy changing of a field and base element as recommended in [58] but without change of p (finding of true primes is computationally difficult [58]). The discrete logarithm calculation algorithm [56] is computationally difficult in the case of $GU(m,p,n)$ for large p since the maximal element order in $GU(m,p,n)$ is p^{n-1}. By increasing m and n, it is possible to cover any message space as far as messages for the proposed cryptosystem and signature scheme are from Z_{mp^n}.

Chapter 7

CONCLUSIONS AND FUTURE RESEARCH

In this thesis we presented two new modifications of Hill cipher based on the use of pseudo-random eigenvalues for one-time key matrix, generated for each plaintext block; The proposed modifications are secure and efficient. Security analysis shows that HCM-EE and HCM-PRE resist the brute-force attack because of large key space; and they also resist the known plaintext-ciphertext attack because of the use of dynamically changing key matrices similar to HCM-NPT. Experiments showed that the proposed HCM-PRE, HCM-PRE are more effective in encryption quality than HCM-NPT, HCM-H, HCM-HMAC and AES in the case of images with large single color areas.

We also presented DH-like matrix protocol based on matrix multiplication and does not use exponentiation as do other known DH protocol modifications. The concept of the proposed protocol is the same as that of DH: it allows two-way arrival at the same common key that is provided by the use of private key matrices commuting each other as conjugates to diagonal invertible matrices. Generation of the public key is made by multiplication with a publicly known zero-determinant matrix. The non-invertibility of this matrix defines the trap-door property of our protocol.

We also presented an extension of Elgamal public key cryptosystem and signature scheme. The proposed extension uses the group *GU(m,p,n)*.

The amount of work devoted to problems of information security is continuously increasing with the proliferation of distribution and transmission of data over networks.

Some trends to be considered further:

1. Propose a parallel algorithm for both HCM-EE and HCM-PRE
2. Study the weaknesses of the RC4 stream cipher
3. Investigate orthogonal matrix and study the ability to use it in the encryption techniques.
4. Investigate the authentication and digital signature problems

REFERENCES

[1] Hill, L. (1929). Cryptography in an Algebraic Alphabet. American Mathematical Monthly, 36(6): pp. 306-312.

[2] Hill, L. (1931). Concerning Certain Linear Transformation Apparatus of Cryptography. American Mathematical Monthly, 38(3): pp. 135-154.

[3] Mahmoud, A. & Chefranov, A. Hill Cipher Modification Based on Pseudo-Random Eigenvalues HCM-PRE. Journal of Applied Mathematics and Information Sciences, To appear.

[4] Mahmoud A. & Chefranov A. (2009). Hill Cipher Modification Based on Eigenvalues HCM-EE. Proc of the 2nd International Conference on Security of Information and Networks (SIN2009) 2009; Gazimagusa (TRNC) North Cyprus, Elci, A., Orgun, M., and Chefranov, A. (Eds.) ACM, New York, USA, pp. 164- 167.

[5] Stallings W. (2006). Cryptography and Network Security Principles and Practices, 4th ed, Prentice Hall: New Jersey.

[6] Mahmoud, A. & Chefranov, A. (2010). Secure Hill Cipher Modifications and Key Exchange Protocol. Proc of 17th IEEE International Conference on Automation, Quality and Testing, Robotics AQTR 2010- THETA 17th, Romania, Cluj-Napoca.

[7] Chefranov, A. & Mahmoud, A. ElGamal Public Key Cryptosystem and Signature Scheme in GU(m,p,n). (2010). Proc. 3rd International Conference on Security of Information and Networks Taganrog, Rostov-on Don, Russia, pp. 164-167.

[8] Daemen, J., & Rijmen, J. (2000). AES Proposal. (http://www.daimi.au.dk/ivan/rijndael.pdf), [last access date is 23.09.2009]

[9] Data Encryption Standard. (1997). Federal Information Processing Standards Publication (FIPS PUB) 46, National Bureau of Standards, Washington, DC. (http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf), [Last access date is 23-09-2009].

[10] Rivest, R. (1995). The RC5 encryption algorithm. Proc of the 2nd Workshop on Fast Software Encryption, Springer, pp. 86-96.

[11] Rivest, R. Shami. A. & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21(2): pp. 120–126.

[12] Elgamal, T. (1985). A public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms, IEEE Trans. on Information Theory, 31(4): pp. 469-473.

[13] Ziedan, I. Fouad, M. & Salem H. (2003). Application of Data Encryption Standard to Bitmap and JPEG Images. Proc 20th National Radio Science Conference (NRSC),

Egypt, pp. 1-16.

[14] Elkamchouchi, H. & Makar, A. (2005). Measuring Encryption Quality of Bitmaps Images with Rijndael and KAMKAR Block Ciphers. Proc. 22nd National Radio Science Conference (NRSC), Egypt, pp. 1-8.

[15] Ismail, A. Amin, M. & Diab, H. (2006). How to Repair the Hill Cipher. Journal of Zhejiang University Science. A, 7(12): pp. 2022-2030.

[16] Saeednia, S. (2000). How to Make the Hill Cipher Secure, Journal of Cryptologia, 24(4): pp. 353-360.

[17] Overbey, J. Traves, W. & Wojdylo, J. (2005). On the Key Space of the Hill Cipher. Journal of Cryptologia, 29(1): pp. 59-72.

[18] Lin, C.H, Lee, C.Y. & Lee, C.Y. (2004). Comments on Saeednia's Improved Scheme for the Hill Cipher. Journal of the Chinese Institute of Engineers, 27(5): pp. 743-746.

[19] Chefranov, A. (2007). Secure Hill Cipher Modification SHC-M. Proc. of the 1st International Conference on Security of Information and Networks (SIN2007) Gazimagusa (TRNC) North Cyprus, Elçi, A., Ors, B., and Preneel, B. (Eds.) Trafford Publishing, Canada, pp. 34-37.

[20] Mohsen, T. & Abolfazl, F. (2011). A Secure Cryptosystem Based on Affine Transformation. Journal of John Wiley & Sons, 4(2): pp. 207-215.

[21] Romero, Y. Garcia R. et al. (2007) Comments on How to Repair the Hill Cipher. Journal of Zhejiang University Science A, 9(2): pp. 211-214.

[22] Li, C. Zhang, D. & Chen G. (2008). Cryptanalysis of an Image Encryption Scheme Based on the Hill Cipher. Journal of Zhejiang University Science A, 9(1): pp. 1118-1123.

[23] Rivest, R. The MD5 Message-Digest Algorithm. Internet RFC 1321, April 1992. Federal Information Processing Standard (FIPS) 180-2. Secure Hash Standard, NIST, U. S 2002, Department of Commerce.

[24] Schneier, B. (1996). Applied cryptography: Protocols, Algorithms, and Source Code in C, 2^{nd} ed, John Wiley & Sons, New York.

[25] Konheim, A. (2007) Computer Security and Cryptography. John Wiley & Sons New Jersey.

[26] Koblitz, N. (1987). A Course in Number Theory and Cryptography. Springer-Verlag, New York.

[27] Galvin, W. (1984). Matrices with Custom-Built Eigenspaces. this MONTHLY, 91:

pp. 308-309.

[28] Apostol, T. (1976). Introduction to Analytic Number Theory. Springer.

[29] Robshow, M. (1995). Stream Ciphers. RSA Laboratories Technical Report TR-701, (http://www.rsasecurity.com/rsalabs/index.html), [last access date is 23.09.2009]

[30] Hossam, E., Ahmed, H. & Osama S. (2007). An Efficient Chaos-Based Feedback Stream Cipher (ECBFSC) for Image Encryption and Decryption. Journal of Computing and Informatics, 31(1): pp. 121-129

[31] Yaobin, M. & Guanrong, C. (2004). Chaos-Based Image Encryption in Eduardo Bayro-Corrochano, editor, Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neural Computing and Robotics. Springer-Verlag, Heidelberg.

[32] Yaobin, M., Guanrong, C. & Shiguo, L. (2006). A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps. Chaos Solitons and Fractals, 21(3): pp. 749-761.

[33] Alvarez, G. & Li, S. (2006) Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems. International Journal of Bifurcation and Chaos, 16(8): pp. 2129–2151.

[34] Hossam, E., Ahmed, H. & Osama, S. (2007). Encryption Effeciency Analysis and Security Evaluation of RC6 Block Ciphers for Digital Images. International Journal of Computer and Information Technology, 1(1): pp. 33-39.

[35] Shannon C.E, Communication Theory of Secrecy Systems. Bell Systems Technical Journal 1948; 28: pp. 656-715.

[36] Cheng, H. & Xiaobo, L. (2000). Partial Encryption of Compressed Images and Videos. IEEE Transaction on Signal Processing, 48(8): pp. 2439-2451.

[37] Marvel, L., Boncelet, G. & Retter C. (1999). Spread Spectrum Image Stegnography. IEEE Transaction Signal Processing, 8(8): pp. 1075-1083.

[38] Lian, S., Sun, J. & Wong, Z. (2008) Security Analysis of Chaos-Based Image Encryption Algorithm. Physics Letters A, 372(15): pp. 2645–2652.

[39] Diffie, W. & Hellman, M. (1976). New Directions in Cryptography. IEEE Transaction on Information Theory, 22(6): pp. 644-654.

[40] Diffie, W., Oorschot, V. & Wiener, M. (1992). Authentication and Authenticated Key Exchange, Designs, Codes, and Cryptography, (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.6682). 107-125, [last access date is 1.11.2010]

[41] Menezes, A. Qu, M.& Vanstone, S. (1995). Some new key agreement protocols providing mutual implicit authentication. The 2nd Workshop on Selected Areas of Cryptography (SAC 95).

[42] Law, L., Menezes, A. Qu, Solinas, J. & Vanstone, S. (2003). An Efficient Protocol for Authenticated Key Agreement, Designs, Codes, and Cryptography, 28(2): pp. 119-134.

[43] Krawczyk, H., 2010. HMQV: A High-Performance Secure Diffie-Hellman Protocol, (http://eprint.iacr.org/2005/176.pdf), [last access date is 5.11.2010].

[44] Odoni, R., Varadharajan, V. & Sanders, P. (1984). Public Key Distribution in Matrix Rings, Electronics Letters, 20(09): 386-387.

[45] Varadharajan, V. & Odini, R. (1986). Security of Public Key Distribution in Matrix rings, Electronics Letters, 22(1): pp. 46-47.

[46] Alvarez, R., Martinez, F., Vicent, J. & Zamora, A. (2008). A Matricial Public Key Cryptosystem With Digital Signature. WSEAS Transaction on Mathematics, 7(4): 195-204.

[47] Alvarez, R. Tortosa, L., Vicent, J.-F. & Zamora, A. (2009). Analysis and Design of A Secure Key Exchange Scheme. Information Sciences, 17(09): pp. 2014-2021.

[48] Nelson, J. (2003). The Diffie-Hellman Key Exchange Protocol in Matrices Over A Field and A ring, MS. Thesis, Dept. Math., Texas Technical University, (http://etd.lib.ttu.edu/theses/available/etd-06262008-31295017074922/unrestricted/31295017074922.pdf), [last access date is 10.11.2010]

[49] Vasco, M. Del Pozo, A. & Duarte, P. (2009). Cryptanalysis of A key Exchange Scheme Based on Block Matrices, 1-16. (http://eprint.iacr.org/2009/553.pdf), [last access date is 10.11.2010]

[50] Yang, J. &Yang, X. (2008). A New Variant of The Diffie-Hellman Key Exchange Protocol Based on Block Triangular Matrix Groups. International Conference of Intelligent Information Hiding and Multimedia Signal Processing, 1277-1287. (http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4604276), [last access date is 22.12.2010]

[51] Anshel, I. Anshel, M. & Goldfeld, D. (1999). An Algebraic Method for Public-Key Cryptography. Mathematical Research Letters, 6: pp. 1-5.

[52] Grigoriev, D. & Ponomarenko, I. (2005). Constructions in Public-Key Cryptography over Matrix Groups, Contemporary Mathematics, AMS, pp. 103- 119 (http://arxiv.org/PS_cache/math/pdf/0506/0506180v1.pdf), [last access date is 1.11.2010]

[53] Elgamal, T. (1985). A Public Key Cryptosystem and A signature Scheme Based on

Discrete Logarithms. IEEE Transaction on Information Theory, 31(4): pp. 469-473.

[54] Pohlic, S. & Hellman, M. (1978). An Improved Algorithm for Computing Logarithms over GF(p) and Its Cryptographic Significance. IEEE Transaction on Information Theory, 24(1): pp. 106-110.

[55] Judson, T (2009). Abstract Algebra. Theory and Applications, Stephen F. Austin State University, February 14, 2009, (http://abstract.ups.edu), [last access date is 25.12.2009]

[56] Dubrois, J. & Dumas, J. (2006). Efficient Polynomial-Time Algorithms Computing Industrial-Strength Primitive Roots, Information Processing Letters, 97(2): pp. 41-45.

[57] Odlyzko, A. (1985). Discrete Logarithms in Finite Fields and Their Cryptographic Significance. Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques, pp. 224 – 314. (https://eprints.kfupm.edu.sa/35266/1/35266.pdf), [last access date is 02.03.2010]

[58] Maurer, U. (1995). Fast generation of Prime Numbers and Secure Public-Key Cryptographic Parameters. Journal of Cryptology, pp. 123-155 (http://reference.kfupm.edu.sa/content/f/a/fast_generation_of_prime_numbers_and_sec_123040.pdf), [last access date is 02.03.2010]

APPENDIX

Appendix:

To demonstrate the results of our experiments, here in, we provide a brief description of image encryption application with screen shots. The application is designed in a flexible way to enable the user to select the image to be encrypted and to save the encrypted image with a given name. It shows both images before and after encryption, decryption, respectively.

The application is written in C# (Microsoft Visual Studio 2008). The choice is dedicated by the fact that, C# offers tools for easy creation of friendly user interfaces; it includes a huge number of functions related to the images and image processing. It is easy to use and compare the results. The encryption/decryption schemes can invoked through a menu including options for the main encryption and decryption algorithms. The options have submenus corresponding to the different algorithms.

The main principle in the system design is its modularity, which makes it extendable. Thus, it is very easy to add algorithms for new encryption algorithms. The system includes a class of methods for encryption and decryption. It also can show (calculate) the quality of encryption by using the quality encryption measures, Irregular Deviation Based Quality Measure (ID) and Correlation Coefficients (C.C).

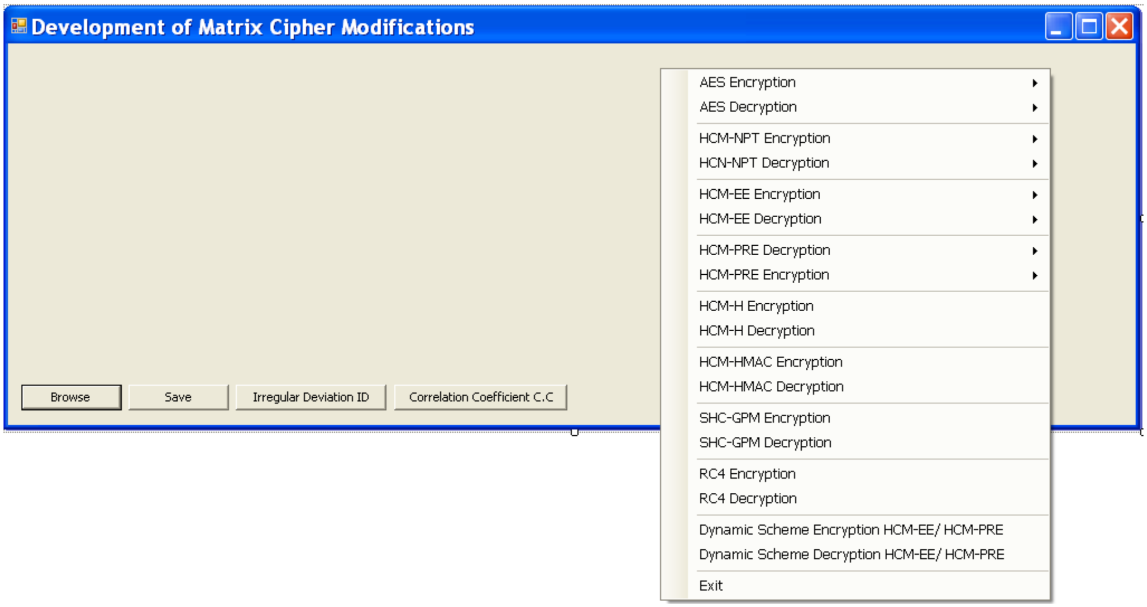
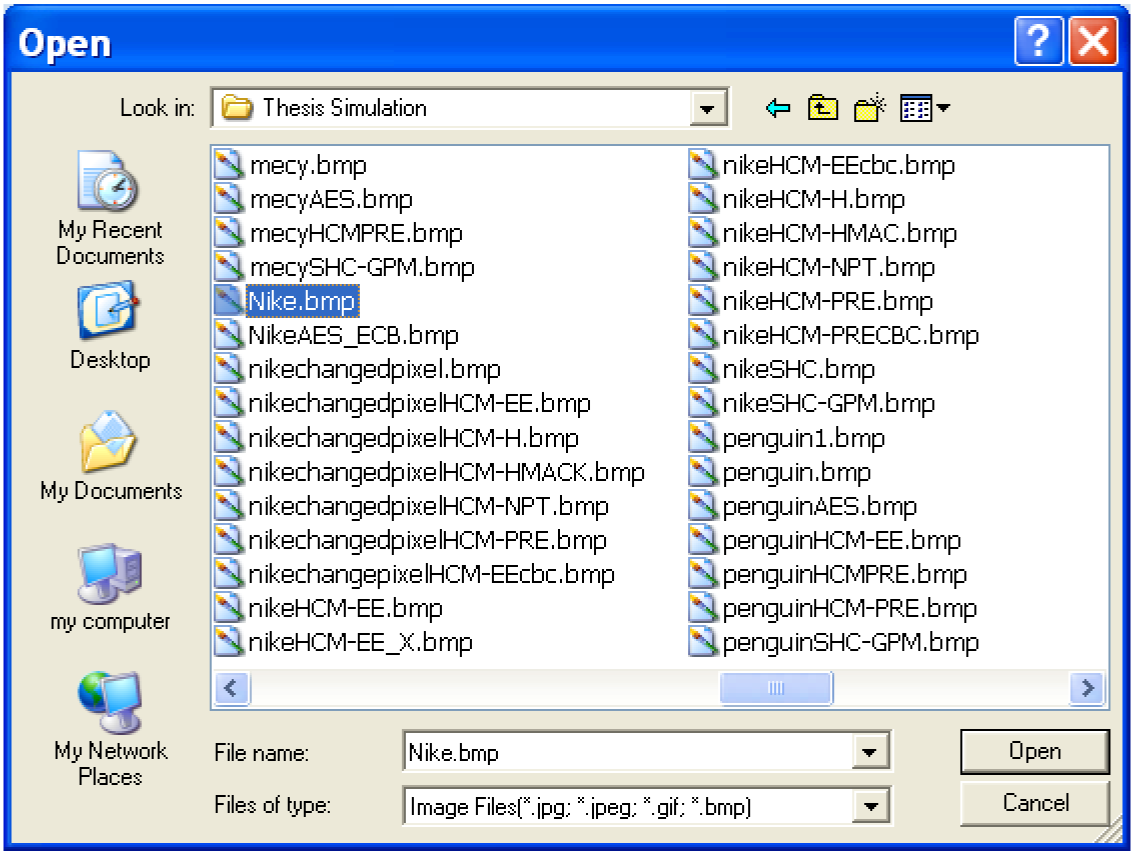


Figure Appendix 1: Main menu of the application

The application can encrypt images with different extensions (types), such as BMP, JPG, GIF, etc. it can be extended easily to cover others. Correspondingly, each encryption scheme includes two submenus, as shown on (Fig. Appendix 1).

To encrypt an image by using our proposed modifications or any other schema, the user first selects the image to be encrypted by clicking on **Browse** (Fig. Appendix 1). The original image to be encrypted is chosen in a flexible way (Fig. Appendix 2). The original image will be displayed as shown in (Fig. Appendix 3, e.g., Nike.bmp). Right click on the mouse to select the schema for encryption (Fig. Appendix 4, e.g., HCM-PRE). The encrypted image will be displayed. The encrypted image can be saved by clicking save button.



Look in: Thesis Simulation

File name: Nike.bmp

Files of type: Image Files(*.jpg; *.jpeg; *.gif; *.bmp)

Figure Appendix 2: Select an image to be encrypted

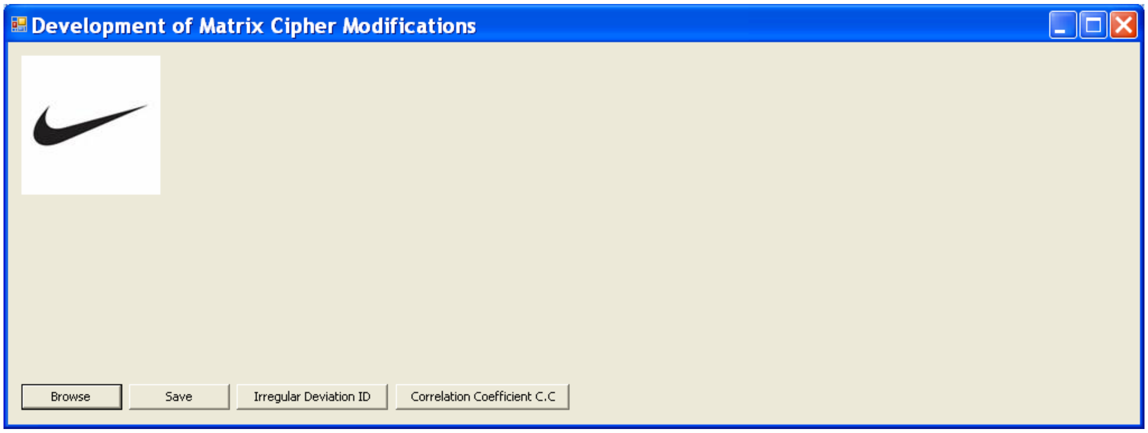

Development of Matrix Cipher Modifications
Browse
Save
Irregular Deviation ID
Correlation Coefficient C.C

Figure Appendix 3: Original image to be encrypted

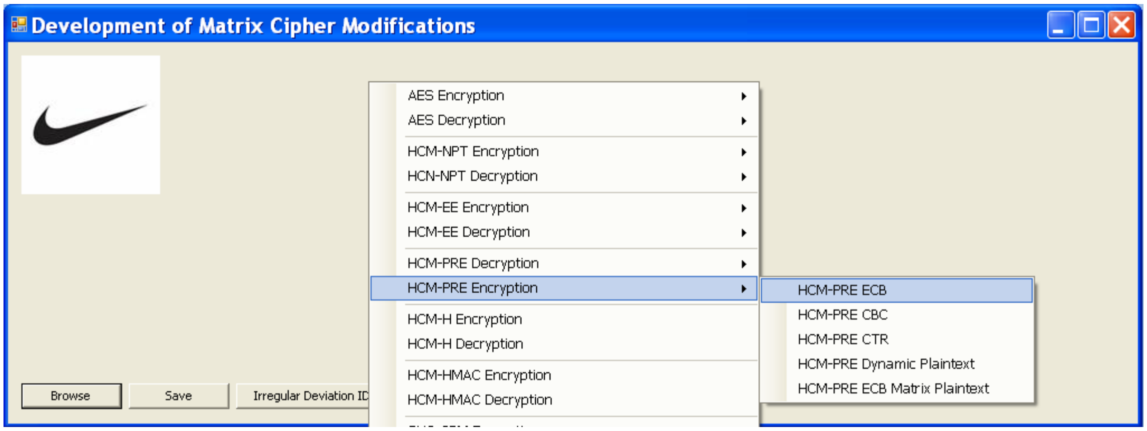


Figure Appendix 4: Nike.bmp will be encrypted by HCM-PRE

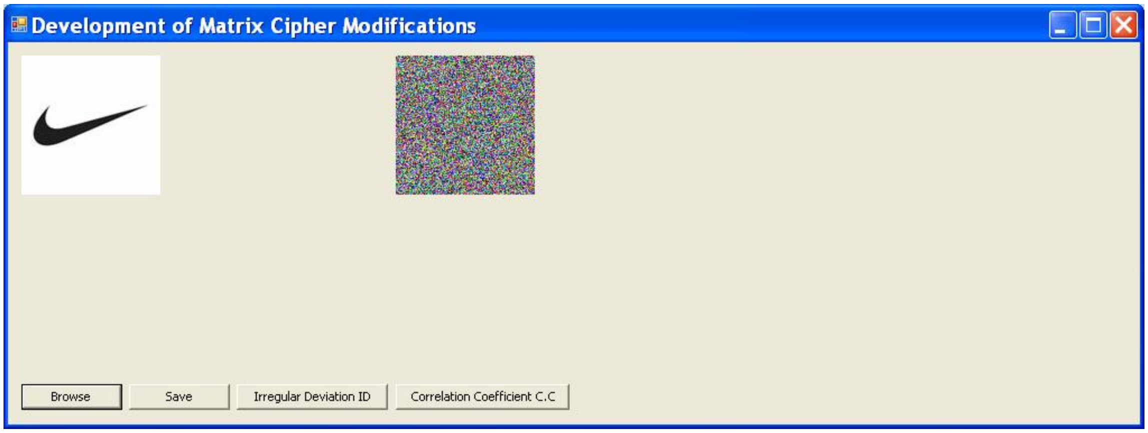
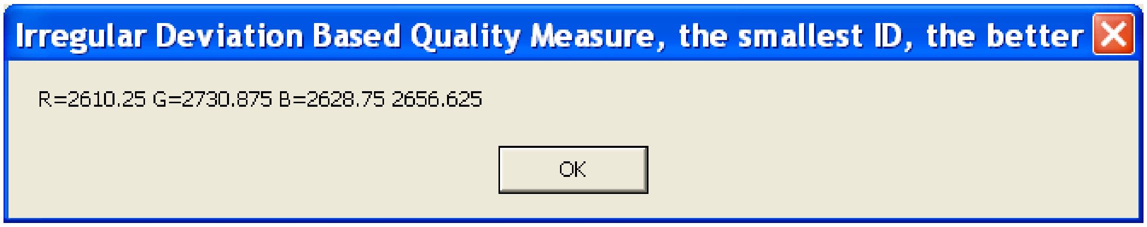

Development of Matrix Cipher Modifications
Browse
Save
Irregular Deviation ID
Correlation Coefficient C.C

Figure Appendix 5: Nike.bmp HCM-PRE encrypted

The irregular irregular deviation based quality measure ID can be calculated by clicking on Irregular Deviation button, the result will be displayed (Fig. Appendix 6), R-value, G-value, and B-value means the ID for red, green, and blue layers, the ID is taken as an average from those three values.


Irregular Deviation Based Quality Measure, the smallest ID, the better

R=2610.25 G=2730.875 B=2628.75 2656.625

Figure Appendix 6: Irregular deviation based quality measure for Nike.bmp

To demonstrate the correctness of our modifications that is, decryption will lead to the original image we perform the HCM-PRE decryption (Fig. Appendix 7)

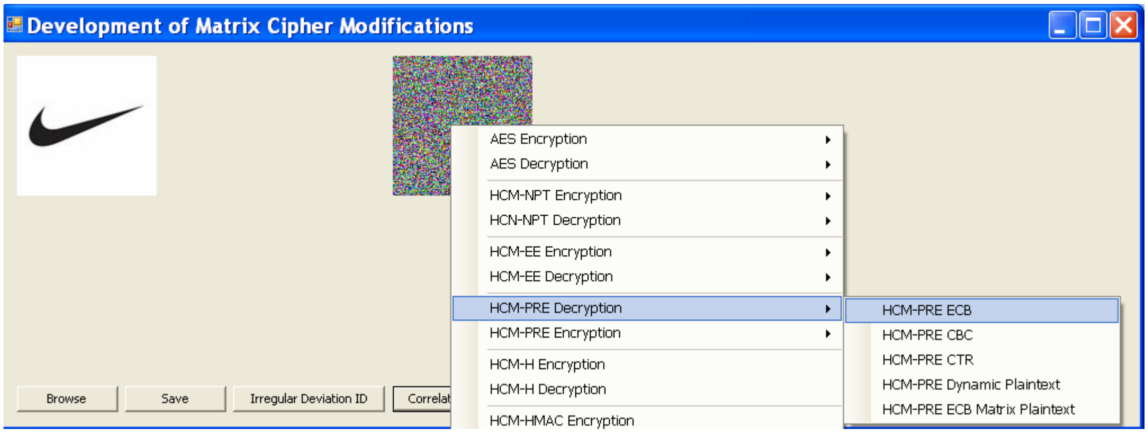


Figure Appendix 7: Nike.bmp will be decrypted by HCM-PRE

The original image is restored (Fig.Appendix 8)

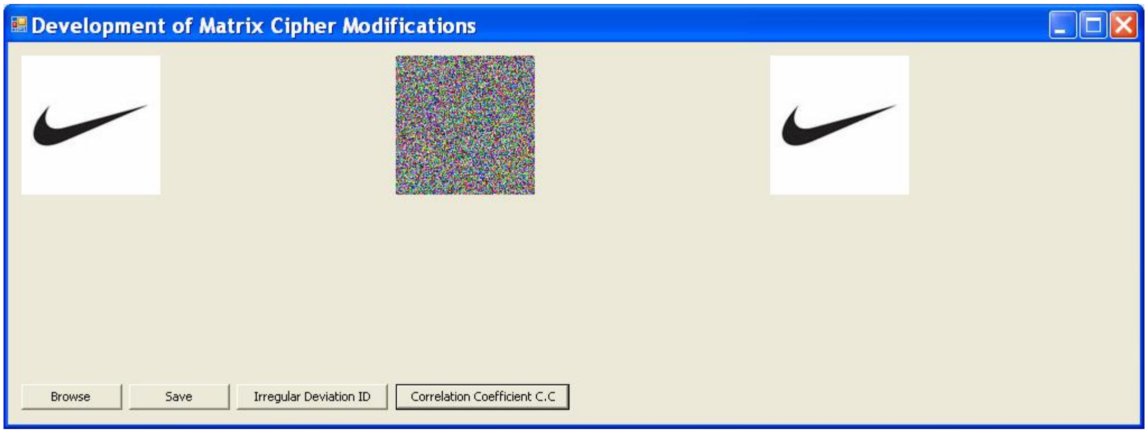

Development of Matrix Cipher Modifications
Browse
Save
Irregular Deviation ID
Correlation Coefficient C.C

Figure Appendix 8: Nike.bmp HCM-PRE decrypted

The former steps can be applied for any encryption schema.

It is appropriate to mention that, we did not include the statistical values of encryption/decryption because of the following: for example, if we used the nike.bmp with size 124x124x3=46128 pixels (bytes) (approximately needs 15 pages with A4), for the irregular deviation based quality, it calculates the histogram of the difference between the original and encrypted image. Hence, this means that we need to include such size of data three times which is not appropriate, therefore, we did not include them.