Title of thesis **A Hybrid Chaotic Image Encryption Scheme Based on S-Box and Ciphertext Feedback**

I, MUHAMMAD ASIM

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP.
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☒ Non-confidential

If this thesis is confidential, please state the reason:

The contents of the thesis will remain confidential for ___________ years.

Remarks on disclosure:

Endorsed by

Signature of Author

Signature of Supervisor

Permanent: *Moh. Baba Kheel, Village: Gujar Garhi*
Address *Tehsil & District: Mardan, Prov: N.W.F.P*
Pakistan

Name of Supervisor
Assoc. Prof. Dr. Varun Jeoti

Date: 23-07-2007

Date: 23-07 2007

UNIVERSITI TEKNOLOGI PETRONAS

Approval by Supervisor

The undersigned certify that they have read, and recommend to The Postgraduate Studies Programme for acceptance, a thesis entitled

A Hybrid Chaotic Image Encryption Scheme
Based on S-box and Ciphertext Feedback

submitted by

Muhammad Asim

for the fulfilment of the requirements for the degree of

Masters of Science in Electrical and Electronic Engineering

23-07-2007
Date

Signature : Varun
Main Supervisor : *Assoc. Prof. Dr. Varun Jeoti*
Date : 23-07-2007
Co-Supervisor :

t
TA
1637
.M952
2007

1) Image Processing -- Security measures
2) Data encryption (Computer science)

UNIVERSITI TEKNOLOGI PETRONAS

A Hybrid Chaotic Image Encryption Scheme Based on S-box and Ciphertext Feedback

By
Muhammad Asim

A THESIS

SUBMITTED TO THE POSTGRADUATE STUDIES PROGRAMME

AS A REQUIREMENT FOR THE

DEGREE OF MASTERS OF SCIENCE IN ELECTRICAL AND ELECTRONIC

ENGINEERING

Electrical and Electronic Engineering

BANDAR SERI ISKANDAR,
PERAK

May, 2007

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Signature:

Name : *Muhammad Asim*

Date : 23-07 2007

ACKNOWLEDGEMENT

First of all I wish to thank ALLAH who provided me the courage and strength that enabled me to complete this work.

I would like to acknowledge the immense contributions, guidance and encouragement from my supervisor Assoc. Prof. Dr. Varun Jeoti. He has been a mentor and also friend to me. Throughout the whole duration of this work, I have learnt much from him, not only in terms of technical knowledge but also the finer points of being a better researcher. Without his guidance, this work would not have yielded such positive results and published at international levels. Indeed his dedication is such that it surpasses boundaries of office hours as evident by the hours of personal time that he has spent with me in discussions of the research work.

I would also like to thank Dr. Shujun Li from Xi'an Jiantong University for very helpful discussions, reviews and comments on our proposed schemes. He is an expert in the field of chaotic cryptography and cryptanalysis but yet he displayed such willingness to help a beginner like me and answered my questions regularly. I feel highly privileged to be able to co-author one paper with him on the security of a recently proposed cipher by Pareek et al. in [Pareek et al., 2006].

I am grateful to the fellow postgraduate students and undergraduate students in UTP that have been through the thick and thin with me throughout the duration of my pursuit for this Master degree. They have been constant sources of encouragement and companionship in times when the going was getting very tough. Their friendships are likened to rare gems.

I would like to pay the most sincere thanks to my parents, whose teaching, encouragement and education, enabled me to pursue achievement in research. I would also like to thank my uncle Mr. Mian Unwan-Ud-Din, from whom I got instruction on each and every step, whenever I got struck. I would also like to thank my dearest friend Mr. Aimal Hayat for his continuous support during my pursuit of this Master degree.

ABSTRACT

The fascinating developments in digital image processing and network communications during the past decade have created a great demand for real-time secure image transmission over the internet and through wireless networks. Due to some intrinsic features of images, such as bulk data capacity and high correlation among pixels, traditional encryption algorithms such as IDEA, DES and AES are not suitable for practical image encryption, especially for real time applications. In order to meet these challenges, a number of schemes have been proposed for encryption of digital images, making use of chaotic dynamical systems.

The objective of the work undertaken in this thesis is two-fold - firstly to evaluate the security of a few representative chaotic ciphers by performing the cryptanalysis on them and secondly, to design an appropriate cipher that would fulfill the needs for both security and speed.

The cryptanalysis is performed on two recently proposed chaotic ciphers by Pareek et al. in [Pareek et al., 2005] and [Pareek et al., 2006]. The first cipher is a generic chaotic block cipher. It is shown that the proposed cipher is insecure against differential and known-plaintext attacks. We also show that the key space size of the proposed cipher is less than what is claimed by the authors. The second cipher of Pareek et al. is a complete image encryption scheme. This scheme is also shown insecure against the differential attack in the thesis. It is also shown suffering from a few security defects and, therefore, is not suitable for real time secure encryption of digital images.

In this work, a complete image encryption scheme - Hybrid Chaotic Image Encryption Scheme (HyChIES) is designed. HyChIES is based on a cryptosystem consisting of multiple piecewise linear chaotic maps (*m*-PLCMs), a generalized logistic map, AES S-box and ciphertext feedback. The analysis of the HyChIES shows that it is extremely sensitive to changes in pixels and, therefore, has an avalanche effect – a highly desirable property for any cipher. As a result, HyChIES randomizes plain images very effectively.

In this thesis, an AES like 128-bit block cipher is also designed, named as Hybrid-Chaotic Encryption Scheme (H-CES). The heart of HyChIES and H-CES is the same cryptosystem that consists of AES S-box, generalized logistic map and ciphertext feedback. In order to analyze the differential characteristic probability of this cryptosystem, we consider it as a hybrid S-box. Based on the maximum differential probability of this hybrid S-box, differential characteristic probability for two rounds of H-CES is calculated and it is shown that H-CES is secure against differential cryptanalysis.

ABSTRAK (BAHASA MALAYSIA)

Perkembangan yang mengagumkan dari pengolahan imej digital dan komunikasi jaringan dalam beberapa dekad terakhir telah menciptakan permintaan yang besar untuk transmisi imej yang selamat dan 'real-time' melalui internet dan jaringan tanpa wayar. Berdasarkan karakteristik 'intrinsic' dari sebuah imej, seperti 'bulk capacity' dan korelasi yang tinggi antara pixel, enkripsi tradisional seperti IDEA, DES dan AES tidak sesuai untuk enkripsi imej yang praktikal, terutama untuk aplikasi real-time. Untuk mengatasi cabaran ini, beberapa skim telah diajukan untuk enkripsi imej digital dengan memanfaatkan system 'chaotic' yang dinamis.

Terdapat 2 objektif dalam kerja yang dilakukan di dalam tesis ini – pertama ialah mengevaluasi keselamatan dari beberapa 'chaotic cipher' dengan melakukan cryptanalysis dan kedua ialah untuk merekacipta sebuah 'cipher' yang memenuhi keperluan keselamatan dan kelajuan.

'Criptanalysis' dilakukan dengan menggunakan dua buah chaotic cipher yang diajukan oleh Pareek [Pareek et al., 2005] [Pareek et al., 2006]. Cipher yang pertama adalah 'blok chaotic cipher' yang umum. Namun, telah dibuktikan bahawa cipher tersebut tidak selamat terhadap serangan yang berbeza dan teks asal yang telah diketahui. Kami juga menunjukkan bahawa ukuran kunci dari cipher yang diajukan lebih kecil daripada yang nyatakan oleh Pareek [Pareek et al., 2005]. Cipher kedua dari Pareek adalah skim enkripsi imej yang lengkap. Namun, skim ini tidak selamat terhadap serangan yang berbeza, seperti ditunjukkan dalam tesis ini. Ianya juga mempunyai beberapa kelemahan dari segi keselamatan dan oleh itu, ia tidak sesuai untuk enkripsi imej digital yang selamat dan 'real-time'.

Pada kerja ini, sebuah skim enkripsi imej yang lengkap-'Hybrid Chaotic Image Encryption Scheme (HyChIES)' telah direkacipta. HyChIES berdasarkan cryptosystem terdiri atas multiple-piece wise linear chaotic maps (*m*-PLCMs), sebuah peta logika yang umum, AES S-box dan umpanbalik ciphertext. Analisis dari HyChIES menunjukkan bahawa ianya sangat sensitif terhadap perubahan pada pixel dan 'avalanche' mempunyai kesan-karakteristik yang sangat di ingini bagi setiap 'cipher' dapat mengubah imej yang asal dengan efektif.

Selain itu, sebuah blok cipher 128-bit yang menyerupai AES juga direkacipta, diberi nama Hybrid-Chaotic Encryption Scheme (H-CES). Bahagian utama bagi HyChIES dan H-CES adalah 'cryptosystem' yang sama, yang terdiri AES S-box, peta logika yang umum dan umpanbalik ciphertext. Untuk menganalisa daripada peluang perbezaan karakteristik cryptosystem, kami menganggapnya sebagai Hybrid S-box. Berdasarkan peluang perbezaan yang paling besar dari hybrid S-box, peluang perbezaan karakteristik dari H-CES telah dihitung dan telah dibuktikan bahawa H-CES selamat terhadap teknik perbezaan crypanalysis.

TABLE OF CONTENTS

Status of thesis i

Approval Page ii

Title Page iii

Declaration iv

Acknowledgement v

Abstract vii

Abstrak (Bahasa Malysia) ix

Table of Contents xi

List of Tables xviii

List of Figures xix

1: Introduction 1

- 1.1 Introduction 1
- 1.2 Brief Introduction to Modern Cryptology 2
- 1.3 Image Encryption Schemes 4
- 1.4 Encryption Algorithms 6
- 1.5 Chaos based Image Encryption 6
  - 1.5.1 Chaos theory 7
    - 1.5.1.1 Chaos and Kneading of a Dough 7
    - 1.5.1.2 Sensitivity to Initial Conditions 8
    - 1.5.1.3 Mixing property 10
    - 1.5.1.4 Ergodicity 12
  - 1.5.2 State of the art in chaotic cryptography 13
- 1.6 Research contribution 13
  - 1.6.1 Cryptanalysis of Recent Chaotic Encryption Schemes... 14
  - 1.6.2 Design of Encryption Schemes 14
- 1.7 Thesis structure 16

2: Multimedia Encryption Review 18
2.1 Introduction 18
2.2 Non Chaos based Multimedia Encryption Schemes 21
2.2.1 Complete Encryption Schemes 21
2.2.2 Selective Encryption Schemes 22
2.2.3 Joint Compression and Encryption Schemes 26
2.3 Chaos based Multimedia Encryption Schemes 28
2.3.1 Complete Encryption Schemes 29
2.3.1.1 Image Encryption Schemes based on 1-D Chaotic Map 29
2.3.1.1 Image Encryption Schemes based on 2-D Chaotic Map 32
2.3.1.1 Image Encryption Schemes based on 3-D Chaotic Map 34
2.3.2 Joint Compression and Encryption Schemes 36
2.3.3 Generic Chaotic Encryption Schemes 37
2.4 Summary 38

3: Cryptanalysis of some Chaos based Encryption Schemes 43
3.1 Introduction 43
3.2 Cryptanalysis 43
3.3 Cryptanalysis of Some Representative Chaotic Ciphers 45
3.3.1 Pareek et al. Chaotic block Cipher 45
3.3.1.1 Step A 46
3.3.1.2 Step B 46
3.3.1.3 Step C 47
3.3.1.4 Step D 47
3.3.1.5 Step E 47
3.3.1.6 Step F 48
3.3.1.7 Step G 50
3.3.8.8 Step H 50
3.3.2 Analysis of Pareek et al.'s Chaotic block Cipher 50
3.3.2.1 Low Sensitivity to Plaintext 51
3.3.2.2 Reduced Key Space Size 52

3.3.2.3 Differential Attack – Proposed Attack 1.............. 53
3.3.2.4 Known Plaintext Attack – Proposed Attack 2......... 58
3.3.3 Fei et al.'s Image Encryption Algorithm......................... 59
3.3.3.1 Brief Description of Fei et al.'s Image Cipher......... 60
3.3.3.2 Sub-Algorithm used for Extracting of ICs from External Secret Key................................... 60
3.3.3.3 Reduced Key Space Size................................ 62
3.4.3.4 Brute Force Attack....................................... 63
3.3.4 Pareek et al.'s Image Encryption Scheme........................ 63
3.3.4.1 Step 1.. 63
3.3.4.2 Step 2.. 64
3.3.4.3 Step 3.. 64
3.3.4.4 Step 4.. 65
3.3.4.5 Step 5.. 66
3.3.5 Analysis of Pareek et al.'s Image Encryption Scheme 66
3.3.5.1 On the Question of its Suitability for Real Time Encryption of Images................................. 66
3.3.5.2 Differential Attack....................................... 67
3.3.5.2.1 Obtaining f_m from Two Ciphered Images.................................. 68
3.3.5.3 Defect in Pareek et al.'s Encryption Scheme......... 74
3.3.5.4 On the Sensitivity to change in pixel................. 76
3.4 Discussion... 76
3.4.1 Positional Significance of each Character in the Secret Key..... 77
3.4.2 Effect of Mapping on Key Space Size............................. 77
3.4.3 Sensitivity to Plaintext.. 78
3.4.4 Chaotic Key Stream Extraction.................................... 78
3.5 Summary... 78

4: Proposed Encryption Schemes **79**
4.1 Introduction... 79

4.2 Background........ 80
  4.2.1 Description of AES........ 80
    4.2.1.1 AES Cipher........ 81
    4.2.1.2 AES Inverse Cipher........ 82
  4.2.2 S-box........ 83
  4.2.3 Inverse S-box........ 85
4.3 HyChIES........ 86
  4.3.1 Piecewise Linear Chaotic Map (PLCM)........ 87
  4.3.2 Generalized Chaotic Logistic Map........ 87
  4.3.3 Initialization of HyChIES........ 88
    4.3.3.1 Mapping Two Dimensional R, G and B Channels into One Dimensional Stream...... 88
    4.3.3.2 Extracting a Unique Initial Condition from External Secret Key........ 88
    4.3.3.3 Generating Chaotic Key Stream for masking 1-D Data Stream........ 90
  4.3.4 HyChIES Encryption Algorithm........ 91
    4.3.4.1 Substitution by S-box........ 91
    4.3.4.2 Masking and Mixing........ 92
  4.3.5 HyChIES Decryption Algorithm........ 94
    4.3.5.1 Inverse of Masking and Mixing........ 94
    4.3.5.2 Inverse Substitution by inverse S-box........ 96
4.4 H-CES........ 96
  4.4.1 Initialization of H-CES........ 96
    4.4.1.1 Extracting a Unique Initial Condition from External Secret Key........ 96
    4.4.1.2 Generating Chaotic Key Stream........ 96
  4.4.2 H-CES Cipher........ 97
    4.4.2.1 Substitution by S-box........ 97
    4.4.2.2 Masking and Mixing........ 98
  4.4.3 H-CES Inverse Cipher........ 98

4.4.3.1 Inverse of Masking and Mixing.................. 99
4.4.3.2 Inverse Substitution by inverse S-box........... 99
4.5 Summary... 100

5: Analysis of the Proposed Encryption Schemes 101
5.1 Introduction.. 101
5.2 Analysis of HyChIES 101
5.2.1 Histogram Analysis 102
5.2.2 Correlation Coefficients Analysis 107
5.2.3 Key Sensitivity Analysis 112
5.2.4 Sensitivity to Plain Image 116
5.2.5 Differential Attack... 117
5.2.6 Key Space Analysis... 120
5.2.6.1 Positional Significance of Each Character in the External Secret Key.......................... 120
5.2.6.2 Key Space Size after the Mapping Process..... 122
5.2.7 Computational Complexity..................................... 125
5.3 Analysis of H-CES 126
5.3.1 Differential and Linear Probabilities 126
5.3.2 Differential Cryptanalysis....................................... 127
5.3.3 Linear Cryptanalysis... 127
5.3.4 Differential and Linear Probabilities of an S-Box......... 127
5.3.5 Maximum Differential and Linear Probabilities of S-box... 128
5.3.6 Active S-box.. 129
5.3.7 Differential Cryptanalysis of H-CES.......................... 129
5.3.7.1 Analysis of the Mapping performed by the Generalized Logistic Map 129
5.3.7.2 Generalized Logistic Map and Ciphertext Feedback.. 130

5.3.7.3 Differential Probability: Generalized Logistic Map and Ciphertext Feedback as an S-box......... 132
5.3.7.4 Differential Probability of the Hybrid S-box........ 133
5.3.7.5 Differential Characteristic Probability of H-CES... 137
5.3.2 Related-Key Attacks... 141
5.3.3 Properties of H-CES 142
5.4 Summary... 145

6: Conclusions.. 147
6.1 Introduction.. 147
6.2 Conclusions.. 147
6.3 Suggested Future Work.. 151

A. Appendix: On Securing CKBA: Ciphertext Feedback based Algorithm...... 152
A.1 Introduction.. 152
A.2 Chaotic Key Based Algorithm.. 152
A.3 Modified- Chaotic Key Based Algorithm............................ 154
A.3.1 M-CKBA Encryption Algorithm.................................. 155
A.3.2 Analysis of M-CKBA.. 159
A.4 Summary... 162

B. Appendix: Suggestions to Improve Initialization Step of HyChIES and H-CES.. 163
B.1 Introduction.. 163
B.2 Initialization of HyChIES... 163
B.2.1 Extracting Unique Initial Conditions (ICs) for 16-PLCMs from External Secret Key... 164
B.3 Initialization of H-CES.. 165
B.4 Summary... 165

References.. 166
Publications.. 177

LIST OF TABLES

Table 3.1 Chaotic systems used in Pareek et al.'s block cipher 49

Table 3.2 Encryption of a specific plaintext '*chaotic*' by Pareek et al.'s cipher.. 57

Table 3.3 Encryption of a specific plaintext '*voltage*' by Pareek et al.'s cipher.. 58

Table 5.1 Correlation coefficient analysis.. 112

Table 5.2 NPCR for a number of images... 117

Table 5.3 ICs for typical external secret keys...................................... 123

Table 5.4 Computational complexity of HyChIES................................ 125

Table 5.5 Result of the analysis from Fig. 5.25.................................... 132

Table 5.6 Probability of typical differentials in case of hybrid S-box......... 136

LIST OF FIGURES

Fig.1.1 An illustration of the encryption scenario 3

Fig.1.2 Kneading of a dough 8

Fig.2.1 Candidate domains for the various encryption algorithms 19

Fig.3.1 The histogram for $\Delta P = P_{2i} - P_{1i}$ 55

Fig.3.2 The histogram for $\Delta C = C_{2i} - C_{1i}$ 55

Fig.3.3 Practical demonstration of loss of positional information in Fei et al.' image cipher 62

Fig.3.4 Intervals and their corresponding operations used in Pareek et al.'s image cipher 65

Fig.3.5 f_{Barbra} and f'_{Barbra} 70

Fig.3.6 f_{Girl} and f'_{Girl} 70

Fig.3.7 f_{House} and f'_{House} 71

Fig.3.8 Recovering Plain Images 71

Fig.3.9 Pseudo code for investigating affect of "*modulo*" operator on Δx 72

Fig.3.10 Histogram of mask image obtained from plain images 73

Fig.3.11 Histogram of mask image obtained from ciphered images 73

Fig.3.12 Enhancing the recovered plain-images $f_{\text{R-Girl}}$ and $f_{\text{R-House}}$, with a 5×5 median filter 74

Fig.4.1 Block Diagram of AES for one round 81

Fig.4.2 Pseudo Code for the AES Cipher 82

Fig.4.3 Pseudo Code for the AES Inverse Cipher 83

Fig.4.4 AES S-box.. 84
Fig.4.5 AES Inverse S-box.. 85

Fig.4.6 Scan pattern for reading the R, G and B- Matrices..................... 88

Fig.5.1 Histogram of the ciphered image 4.1.01.tiff............................ 103

Fig.5.2 Histogram of the ciphered image 4.1.02.tiff............................ 103

Fig.5.3 Histogram of the ciphered image 4.1.03.tiff............................ 104

Fig.5.4 Histogram of R-channel of the plain image 4.1.03.tiff................ 104

Fig.5.5 Histogram of R-channel of the corresponding ciphered image 4.1.03.tiff.. 105

Fig.5.6 Histogram of G-channel of the plain image 4.1.03.tiff................ 105

Fig.5.7 Histogram of G-channel of the corresponding ciphered image 4.1.03.tiff.. 106

Fig.5.8 Histogram of B-channel of the plain image 4.1.03.tiff................ 106

Fig.5.9 Histogram of B-channel of the corresponding ciphered image 4.1.03.tiff.. 107

Fig.5.10 Horizontal correlation for plain R-channel 109

Fig.5.11 Horizontal correlation for ciphered R-channel.......................... 109

Fig.5.12 Horizontal correlation for plain G-channel 110

Fig.5.13 Horizontal correlation for ciphered G-channel.......................... 110

Fig.5.14 Horizontal correlation for plain B-channel 111

Fig.5.15 Horizontal correlation for ciphered B-channel.......................... 111

Fig.5.16 Plain image 4.1.01.tiff.. 113

Fig.5.17 Ciphered image of 4.1.01.tiff... 114

Fig.5.18 Decrypted image with wrong key '*abcdefghijklmnoo*'............... 114

Fig.5.19 Histogram of image decrypted with wrong key '*abcdefghijklmnoo*' 114

Fig.5.20 Decrypted image with wrong key '*abcdefghijklmnoo*' 115

Fig.5.21 Histogram of image decrypted with wrong key '*abcdefghijklmnoo*' 115

Fig.5.22 Differential Attack 118

Fig.5.23 Histogram of mask image obtained from plain images 119

Fig.5.24 Histogram of mask image f_m obtained from cipher images 119

Fig.5.25 10000-bar histogram of K_i that consists of 100000 values 124

Fig.5.26 10000-bar histogram of IC after mapping 100000 external keys.... 124

Fig.5.27 Mapping performed by the generalized logistic map 130

Fig.5.28 Mapping performed by the generalized logistic map and ciphertext feedback 131

Fig.5.29 Distribution of differentials for generalized logistic map and ciphertext feedback as S-box 133

Fig.5.30 Distribution of differentials for AES S-box 135

Fig.5.31 Distribution of differentials for Hybrid S-box 135

Fig.5.32 Diffusion property of H-CES 143

Fig.5.33 H-CES Sensitivity to plaintext 144

Fig.5.34 H-CES Key sensitivity 144

Fig.A.1 f :Plain image 5.1.14.tiff 159

Fig.A.2 f':Corresponding cipher image of 5.1.14.tiff 159

Fig.A.3 f_m :Mask image obtained from f and f' 160

Fig.A.4 f':Corresponding cipher image of 5.1.10.tiff 160

Fig.A.5 f_d :Decrypted image using the mask image f_m 161

Fig.A.6 Histogram of the decrypted image f_d, using the mask image f_m .. 161

1 Introduction

CHAPTER

1.1 INTRODUCTION

Currently, many digital services like pay-TV, confidential video conferencing, medical and military imaging systems, require reliable security in storage and transmission of digital images/videos. The prevalence of multimedia technology and, at the same time, rapid progress and wide acceptance of Internet in our society today warrant, on one hand, high security of digital images/videos, and, on the other, protection of users' privacy. To fulfill such security and privacy needs in various applications, encryption of images and videos is very important to frustrate malicious attacks from unauthorized parties.

From early 1990s, many efforts have been made to investigate specific solutions to image/video encryption. To start a comprehensive discussion of encryption and the chaos based algorithms, first a brief introduction to modern cryptology is given, followed by image encryption schemes, and use of chaos in image encryption and the state of the art in chaotic schemes for image encryption. Next, we include contributions made in this thesis, followed by organization of the thesis.

1.2 Brief Introduction to Modern Cryptology

In this section a brief introduction to the modern cryptology is given in order to facilitate the discussion in this dissertation.

Modern *Cryptology* is the technology of encryption that covers the two main areas: 1) cryptography; and, 2) cryptanalysis. *Cryptography* is the science of how to design good encryption algorithms for converting information from its normal, comprehensible form into an incomprehensible form, rendering it unreadable without the secret key. The process of obscuring information to make it unreadable without secret key is called *encryption*, while *decryption* is the process of converting back the obscured information into a readable format with a secret key. The algorithm used during the process of encryption/decryption is called *cipher*. Usually the encryption algorithm is called *encipher*, while the decryption algorithm is called *decipher*. The input to encipher is called the *plaintext* while the input to decipher is called the *ciphertext*, which is the output of encipher. Encryption/decryption algorithm operates on the secret key, usually denoted by K. The details of the encryption/ decryption algorithms are known to the unauthorized user, however he/she does not know the secret key. While, historically, cryptography was used mostly in military settings, it has now become a leading enabler in everyday modern life. From secure purchasing over the Internet to digital satellite decoders, from encrypting cellular conversations to automatic toll collection, from smart cards to electronic voting, cryptography is an essential component.

Cryptanalysis is the science that evaluates the promises of cryptography: When we make a purchase over the Internet, we would like our transaction to be "secure". We wish that attackers would not be able to tap into our personal information, change the results of electronic voting without being detected, make fraudulent calls on our account, etc. Cryptanalysis focuses on evaluating the strength of cryptographic primitives and protocols. It is commonly believed that the fate of an entire nation is affected by cryptography. One example is the German Enigma machine. It started as a commercial cipher, and later continuously improved and used by the German army. The Polish broke the Enigma in the 1930's, and improved their methods side by side with the German improvement of the Enigma. A few weeks before the start of World War II the Polish transferred their knowledge to the French and to the British. The British further improved the Polish methods, and created a huge intelligence organization for deciphering German encrypted communications. Later, the British shared their information with the US forces. Many believe that the allies' ability to decrypt German communication had an overwhelming role in the result of the war.

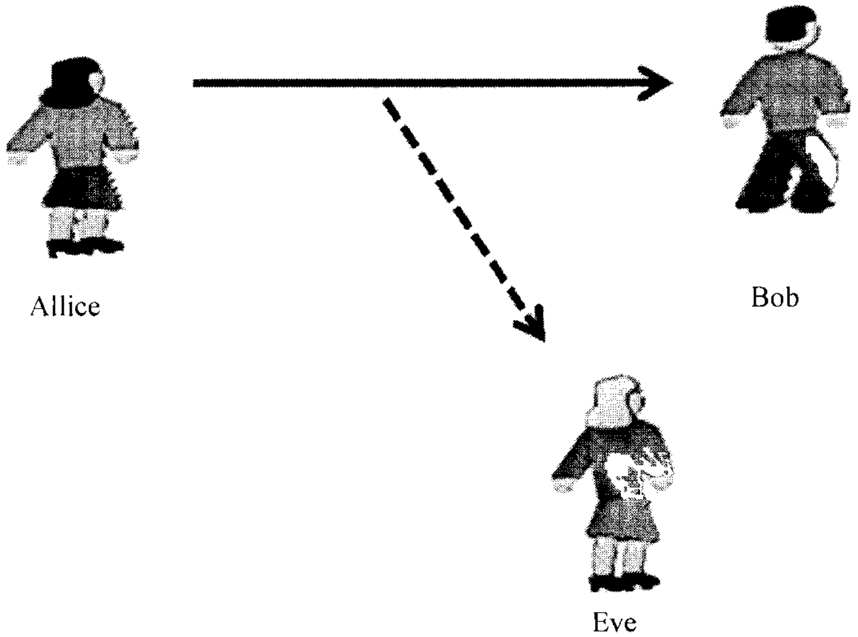


Fig. 1.1. An illustration of the encryption scenario [http://ranger.uta.edu/~nzhang/6392/lectures/4.pdf]

In Fig. 1.1, a typical encryption scenario has been illustrated. Alice wishes to transmit a message (plain image) to Bob. Alice encrypts the message P using the secret key K_e and the encryption rule $C = E(P, K_e)$, where K_e is the encryption key. She then transmits the resulting ciphertext C (ciphered image) to Bob. Bob decodes the message using the received ciphertext C and the key K_d and the decoding rule $P = E^{-1}(C, K_d)$, where K_d is the decryption key.

There are two kinds of ciphers following the relationship of K_e and K_d. When $K_e = K_d$ the cipher is called a private-key cipher or a symmetric cipher. For private-key ciphers, the encryption-decryption key must be transmitted from the sender to the receiver via a separate secret channel. When $K_e \neq K_d$, the cipher is called a public-key cipher or an asymmetric cipher. For public-key ciphers, the encryption key K_e is published, and the decryption key K_d is kept private, for which no additional secret channel is needed for key transfer. The public – key ciphers are usually used for carrying the secret key K for the symmetric or private- key ciphers.

In Fig. 1.1, Eve, who is an unauthorized receiver, attempts to recover the message P and the encryption key K by using the current ciphertext C, and the plaintext P and corresponding ciphertext from other transmissions.

1.3 Image Encryption Schemes

In this thesis, the focus in on image encryption and hence from hereon, the discussion will be limited to the encryption specially designed for images.

Digital images and videos are generally bulky. Since encryption speed of traditional ciphers is not sufficiently fast, it is difficult to achieve real-time secure encryption. Also in digital images, there exists a strong correlation among pixels. The main obstacle in designing image encryption algorithms is that it is rather difficult to swiftly shuffle and diffuse data by traditional means of cryptology. In this respect, chaos-based algorithms have shown their superior performance.

Also due to high redundancy in uncompressed images in the form of high correlation among neighboring pixels, traditional block ciphers fail to conceal all visible information in some plain-images. However, block ciphers running in cipher block chaining mode or what is understood to be 'stream cipher mode' should be used to encrypt uncompressed images with high redundancy. Still the computational burden of traditional ciphers is considerably high.

In addition, traditional ciphers can not provide avalanche property at the level of image. For example if two images differ in only single pixel position, then encryption of two plain-images should result in totally different ciphered images. In order, to maintain the said avalanche property, hence, special algorithm needs to be developed for images. In the current literature image encryption schemes are classified into three main categories: 1) complete encryption schemes; 2) selective encryption schemes; 3) joint compression-encryption schemes.

Complete encryption scheme works in spatial domain and encrypts the image before or after lossy or lossless compression, while selective and joint compression-encryption schemes work in the transform domain. Using complete encryption approach, different types of ciphers have been proposed for the encryption of digital images such as DES [Dang & Chau, 2000] [Ziedan et al., 2003], AES [Hou & Wang, 2003], and FEA-M [Yi et al., 2001]. Even though, complete encryption schemes like DES and AES have high security, they are computationally very intensive. It should also be mentioned that cryptanalysis of FEA-M has been carried out with very low computational load in [Youssef & Tavares, 2003].

Selective encryption schemes only encrypt the data stream that carries the rich content. This approach has gained considerable attention of the researchers, aiming to achieve a better tradeoff between the encryption load and the security level. Different schemes have been proposed in [Qiao & Nahrstedt, 1997] [Kunkelmann & Reinema, 1997].

In joint compression – encryption schemes, the encryption process is combined with the compression process. Various schemes have been proposed recently in [Lian & Wang, 2003] [Qiao et al., 1997].

Although many efforts have been devoted to better solutions for image and video encryption, the current security analyses of many schemes are not sufficient, especially on the security against known/chosen-plaintext attack. What's worse, many selective encryption schemes have been reported to be insecure against ciphertext-only attack. The detailed discussion of the various schemes is reported in the chapter 2 of this dissertation.

1.4 Encryption Algorithms

Multimedia data has been encrypted using variety of mathematical algorithms, which include – factoring of large prime numbers, finite number theory etc. AES, DES are based on the finite number theory. Most of these algorithms have inherent tradeoff between security and complexity. The search for an efficient algorithm that leads to least complexity for a given security level is still on. Attempts have been made to explore the use of chaos and many other fields to this end. It is believed that chaos based cryptography is simpler to implement compared to most other schemes. In this thesis, an attempt has been made to search for an image encryption schemes that is primarily based on chaos and has AES equivalent security. In the light of the above, it would be appropriate to introduce chaos and its use in encryption.

1.5 Chaos Based Image Encryption

In order to facilitate the future discussion, here chaos theory and the terms related to it are briefly introduced.

1.5.1 Chaos theory

Chaos theory describes the behavior of nonlinear dynamical systems that under certain conditions exhibit dynamics that are extremely sensitive to initial conditions. As a result of this sensitivity, the behavior of chaotic systems appears to be random, as different initial conditions give rise to completely different dynamics.

For a dynamical system to be classified as chaotic, most of the researchers agree that it must have the following three properties: 1) sensitivity to initial condition determined by positive Lyapunov exponent; 2) mixing property; and, 3) ergodicity. The detailed description of the sensitivity to initial conditions, mixing property, and ergodicity is given below.

1.5.1.1 Chaos and Kneading of a Dough

In order to understand the properties of chaos, we take the help of one the most important metaphors in chaos: the kneading of a dough. The kneading process guarantees that a pocket of spices inserted into the dough will be mixed thoroughly throughout the mass. The kneading of dough provides an intuitive access to the above mentioned properties of chaos. Basically there is nothing random about the kneading process itself. Rather, a baker applies a certain action over and over again. The kneading process can be imagined as stretching the dough and folding it over, repeated many times. But in spite of this deterministic definition, the results have many features in common with randomness.

In order to show the randomness, we can take help of Fig. 1.2. Imagine that a dough consists of infinitely thin layers. Folding these layers does not change the thickness and we can represent the dough by a line segment. Fig.1.2 shows some stages of kneading process. We mark two grains of spice and follow their paths. Two grains are rather close together initially. But where will they be after a dozen kneading? It is very likely that we will find them in a very different place in the dough. In fact, that would be a consequence of the mixing properties of kneading. In other words, kneading destroys *neighborhood relations*. Grains which are very close initially will likely not be close neighbors after a while. This is the effect of *sensitive dependence on initial conditions*.

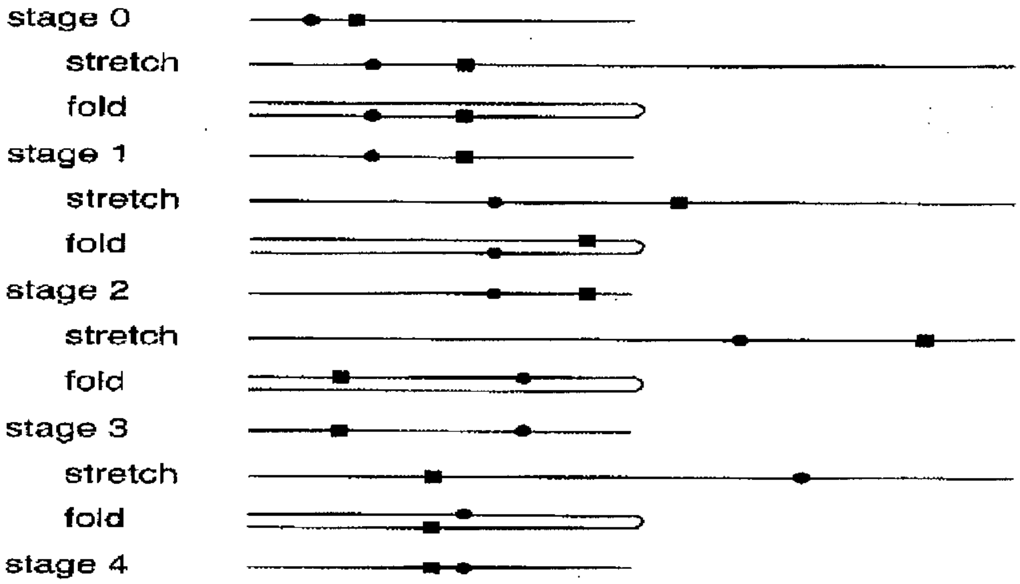


Fig. 1.2. Kneading of a dough: Two grains, symbolized by a dot and a square, are subjected to four stages of the stretch-and-fold. They are mixed throughout the dough [Peitgen et al., 2004]

1.5.1.2 Sensitivity to Initial Conditions

One of the desirable properties of chaotic dynamical systems that is considered vital for cryptographic algorithms is sensitivity to initial conditions and control parameters. Sensitivity to initial conditions means that when a chaotic map is iteratively applied to two initially close points of data, the points quickly diverge, and eventually become uncorrelated. Sensitivity to parameters causes the properties of the map to change quickly when the parameters on which the map depends are slightly perturbed. Sensitivity to initial conditions is studied by Lyapunov Exponent, explanation of which is given below.

The Lyapunov Exponent

Chaotic dynamics is characterized by an exponential divergence of initially close points. For introduction purposes, here one-dimensional discrete chaotic maps on the interval [0, 1] will be discussed. Consider a nonlinear dynamical system, described by:

$$x_{n+1} = f(x_n), \quad x \in [0,1] \tag{1.1}$$

where f is a nonlinear function.

The so called '*Lyapunov exponent*' is a measure of the divergence of two orbits starting with slightly different initial conditions x_0 and $x_0 + \Delta x_0$. The distance after n iterations given by (1.2)

$$\Delta x_n = \left| f^n(x_0 + \Delta x_0) - f^n(x_0) \right| \tag{1.2}$$

increases exponentially for large n for a chaotic orbit according to

$$\Delta x_n \approx \Delta x_0 e^{\lambda_L n} \tag{1.3}$$

where λ_L is the Lyapunov exponent.

One can now relate the Lyapunov exponent analytically to the average stretching along the orbit $x_0, x_1 = f(x_0), x_2 = f(f(x_0)), \cdots, x_n = f^n(x_0) = f(f(f...(x_0)...))$. From (1.2) and the chain rule of differentiation, we have

$$\begin{aligned}\ln\frac{\Delta x_n}{\Delta x_0} &\approx \ln\left|\frac{f^n(x_0+\Delta x_0)-f^n(x_0)}{\Delta x_0}\right| \\ &\approx \ln\left|\frac{df^n(x)}{dx}\right| = \ln\prod_{j=0}^{n-1}\left|f'(x_j)\right| \\ &= \sum_{j=0}^{n-1}\ln\left|f'(x_j)\right| \end{aligned} \tag{1.4}$$

From (1.3), Lyapunov exponent can be written as:

$$\begin{aligned}\lambda_L &= \lim_{n\to\infty}\frac{1}{n}\ln\frac{\Delta x_n}{\Delta x_0} \\ &= \lim_{n\to\infty}\frac{1}{n}\sum_{j=0}^{n-1}\ln\left|f'(x_j)\right| \end{aligned} \tag{1.5}$$

where the logarithm of the linearized map is averaged over the orbit $x_0, x_1, x_2, \ldots\ldots, x_n$.

Negative values of the Lyapunov exponent indicate stability, and positive values chaotic evolution, where λ_L measures the speed of exponential divergence of neighboring trajectories. In other words, Lyapunov exponent provides a measure for the rates of sensitivity for a chaotic system to the two nearby initial conditions and control parameters.

1.5.1.3 Mixing Property

Mixing is the tendency of the chaotic systems to quickly "scramble up" small portions of the state space into an intricate network of filaments. Local, correlated information becomes scattered all over the state space. Mixing is regarded as one of the essential signs for the chaotic attributes for a chaotic system. In discussion of sensitivity to initial conditions and Lyapunov exponent, we mentioned that small error is multiplied exponentially in the course of iteration. Now we will interpret the same thing from a slightly different point of view.

A bit more formally, the mixing property is defined in the following way [Peitgen et al., 2004]:

- *For any two open intervals I and J (which can be arbitrarily small, but must have a nonzero length) one can find initial values in I which, when iterated, will eventually lead to points in J*

Mixing property is regarded as an essential source of diffusion in the encryption algorithms. In [Kocarev et al., 1998], authors quotes from [Shannon, 1949] as:

> "*... we can think of a mixing transformation as one which distributes any reasonably cohesive region in the space fairly uniformly over the entire space. If the first region could be described in simple terms, the second would require very complex ones... If a mixing transformation were applied, the high probability messages would be scattered evenly throughout the space*".

Researchers believe that the mixing property of chaotic maps is closely related to property of diffusion in encryption algorithms. If the set of all the possible plaintexts are considered as initial region in the phase space of the map (transformation), then it is the mixing property (or in other terms, sensitivity to initial conditions) that implies "spreading out of the influence of a single plaintext digit over many ciphertext digits". On the other hand, a good encryption algorithm spreads also the influence of a single key digit over many digits of ciphertext. The keys of an encryption algorithm represent its parameters. Therefore, the transformations in which both parameters and variables are involved in a sensitive way should be considered i.e. "a small variation of any one" (variable, parameter) "changes the outputs considerably".

1.5.1.4 Ergodicity

Definition of ergodicity was first given by Boltzman. A system which is thoroughly stirred is said to be *ergodic*. The Boltzman definition of ergodicity is given as:

> *"In an ergodic system the trajectory of almost every point in phase space eventually passes arbitrarily close to every other point on the whole interval"*

As a direct consequence of such property, the time averages are equal to space averages. Hence for any integrable function f,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} f(T^k x) = \int_X f du. \tag{1.6}$$

In (1.6), the time average of f for n time steps along the orbit of x is represented by averaging $f(x)$ with $f(Tx)$, $f(T^2x),\ldots,$ and $f(T^{n-1}x)$, since each application of T represents the passage of one unit of time (the left hand side of the expression). The space average of f is obtained by integrating f over the entire space X, giving the right hand side of the expression.

Ergodicity is an essential property for chaotic systems and is closely related to the mixing property. For an ergodic map, the trajectory or orbit of each initial point from its definition interval will lead to the same distribution after a large number of iterations. Ergodicity means that if one picks a number x_0 and another number x_1 at random, then it is always possible to get to x_1 during the course of iterations sometime or the other. These orbits describing iterations of this nature are called ergodic. Hence when iterated, the ergodic orbits are mixed throughout the whole interval.

1.5.2 State of Art in Chaotic Cryptography

These above mentioned characteristic of the chaotic nonlinear dynamical systems can be linked to "confusion" and "diffusion" property in cryptography. So it is a natural to use chaos to enrich the design of new ciphers. Encouraged by such properties many chaotic ciphers have been proposed and analyzed since 1989.

In the last decade various chaos based image encryption schemes have been proposed. The proposed schemes are based mainly on 1-D, 2-D or 3-D chaotic or hybrid schemes consisting of different dimensional chaotic maps. 1-D chaotic maps are used to generate pseudo-random bits with desired statistical properties [Pareek et al., 2005] [Pareek et al., 2006]. 2-D chaotic maps are mainly used to permute the pixel values dynamically [Chen et al, 2004], while 3-D chaotic maps are used for generating three different sequences of pseudo-random bits with desired statistical properties for the encryption of the R, G and B channels of digital images [Fei et al, 2005].

But the security of the most of chaotic schemes against the known-plaintext and chosen-plaintext attack is still a major concern. The security of chaotic cryptosystems often relies on apparent properties of the chaos. Encryption and cryptanalysis using chaotic dynamics is a relatively new field. Most of chaos based cryptosystems have been broken. Today even their speed has also come under great scrutiny as their throughput proves to be less than that of AES, DES.

1.6 Research Contribution

Our research on the chaotic ciphers was first motivated by the security of the multimedia data, where the volume of the data is huge and conventional cryptographic ciphers are not suitable due to their high computational complexity. It is found that chaotic ciphers are one of the alternatives for the security of the multimedia data considered by the recent researchers. But with the passage of time, it is found that most of the chaotic ciphers are not secure against the known-plaintext and chosen-plaintext attack. Hence the objective of this work was to analyze some of the recent representative chaotic image encryption schemes, learn their positive and negative characteristics and propose more efficient scheme for the encryption of digital images. Hence our contribution can be divided into two parts: 1) Cryptanalysis of recent chaotic encryption schemes; 2) Design of encryption schemes.

1.6.1 Cryptanalysis of Recent Chaotic Encryption Schemes

The cryptanalysis section consists of the following contributions, whose detailed description is given in chapter 3 of this dissertation.

- The cryptanalysis of Pareek et al. chaotic block cipher [Pareek et al., 2005] is performed. It is shown that the proposed cipher is insecure against differential and known plaintext- attack. It is also identified that the cipher has a very low key space size.

- We also identify that an image encryption scheme proposed by Fei et al. in [Fei et al., 2005] similarly has a low key space size and is insecure against the brute force attack.

- We also perform cryptanalysis of a very recently proposed chaotic image encryption scheme by Pareek et al. in [Pareek et al., 2006]. It is shown that the proposed scheme is insecure against the differential attack. We also identify some other security defects in the proposed scheme and highlights that the proposed scheme is not suitable for real time encryption of digital images.

1.6.2 Design of Encryption Schemes

The design section consists of the following contributions:

- We propose an image encryption scheme, named as Hybrid Chaotic Image Encryption Scheme (HyChIES). The proposed scheme is based on multiple-piecewise linear chaotic maps (*m*-PLCMs), a generalized logistic map, AES S-box and ciphertext feedback. In HyChIES, an external secret key of 128 bit is used, further mapped through a sub-algorithm to acquire initial conditions for *m*-PLCMs, used for extracting the key stream for masking the gray level values of the image. Each pixel value is substituted by S-box. Generalized logistic map is used for mixing purpose, whose operands are previous ciphertext pixels (ciphertext feedback at the level of pixels) and corresponding 8-bit key stream extracted from *m*-PLCMs. The analysis of the HyChIES shows that it is extremely sensitive to the change in pixels and fulfills the avalanche property at the level of image and effectively randomizes plain images.

- We also proposed an AES like 128-bit block cipher, named as Hybrid-Chaotic Encryption Scheme (H-CES). The heart of HyChIES and H-CES are similar, it consists of AES S-box, generalized logistic map and ciphertext feedback. In order to analyze the differential characteristic probability for H-CES, the AES S-box, generalized logistic map and the ciphertext feedback are considered as one Hybrid S-box. Based on the maximum differential probability of the Hybrid S-box, differential characteristic probability for two rounds of H-CES is calculated and is proved that H-CES is secure against differential cryptanalysis technique.

Detailed description and analysis of the proposed schemes are given in chapter 4 and 5 of this dissertation.

1.7 Thesis Structure

This thesis is organized as follows: In Chapter 2, a detailed survey of existing encryption schemes for multimedia data e.g. image, video and audio, will be presented. The encryption schemes are classified into the following two major categories: 1) Non chaos based multimedia encryption schemes; 2) Chaos based multimedia encryption schemes. The schemes under each of the two categories are further classified into the following three sub-categories: 1) Complete encryption schemes; 2) Selective or partial encryption schemes; 3) Joint image compression and encryption schemes.

Chapter 3 presents the analysis of a few recently proposed chaotic encryption schemes proposed in [Pareek et al., 2005] [Fei et al., 2005] [Pareek et al., 2006]. It is identified that chaotic systems are mainly used for masking the plain data. However, the secret key stream used for masking the plain data can be extracted using the known plaintext attack. The extracted secret key stream can then be used as equivalent key for recovering another list of plain data [Pareek et al., 2005]. Also it is identified that the key space size of the Pareek et al. and Fei et al.'s cipher is less then 2^{128}, contrary to what is claimed by the authors in [Pareek et al., 2005] [Fei et al.'s 2005].

Chapter 4 describes the proposed image encryption schemes based on generalized logistic map, piecewise linear chaotic maps (PLCMs) and AES S-box. The generalized logistic map operates on the input values between 0 & 255 and works similar to an S-box. However due to its "*two to one*" nature, it does not effectively distributes the differentials, that leads to its high differential probability. The generalized logistic map is only used for the mixing purpose in the proposed schemes and operates on the previous ciphertext byte. The proposed schemes are named as Hybrid Chaotic Image Encryption Schemes (HyChIES) and Hybrid Chaotic Encryption Schemes (H-CES), where HyChIES is a complete image encryption scheme that works on the whole stream of image, while H-CES works on 128 bit block of plain data.

Chapter 5 presents the extensive analysis of the HyChIES and H-CES. It is proved that HyChIES effectively randomizes the input plain images and correlation between a plain and cipher image is negligible. HyChIES fulfills the avalanche property at the level of image and the affect of change in one pixel is distributed over the rest of pixels with in two rounds of encryption. The study of H-CES is followed by the analysis of HyChIES. First the components of H-CES are analyzed, followed by the proof that H-CES is secure against the differential cryptanalysis.

The thesis ends with the conclusion of the work and recommendation for future work in Chapter 6.

In the Appendix A of this thesis, it is shown that the original CKBA proposed by Cheng et al. in [Cheng & Guo, 2000] can be made secure against the proposed attack by Li et al. in [Li & Zheng, 2002 (b)], by using ciphertext feedback and encrypting for multiple number of rounds.

In the Appendix B of this thesis, we suggest modifications to the initialization step of HyChIES and H-CES, in order to make it more convenient and less complex for implementation.

2

CHAPTER

Multimedia Encryption Review

2.1 Introduction

In this chapter a comprehensive literature survey of multimedia encryption schemes is presented. First a detailed survey of non-chaos based multimedia encryption schemes is presented followed by the prevailing chaos based encryption schemes. The goal of the survey is to get a measure of the positive and negative characteristics of the presented schemes. The chapter ends with lessons learnt from the carried survey.

Many approaches have been proposed by researchers for the encryption of multimedia data addressing various issues such as real time constraints, complexity, energy consumption, compression ratio, bit errors, ease of implementation etc. In the current literature, multimedia encryption schemes are divided into the following three categories [Wu & Mao, 2002].

1) Complete encryption schemes
2) Selective/ Partial encryption schemes
3) Joint compression-encryption schemes

Generally there are two basic ways to encrypt the multimedia: in spatial domain or in transform domain. Complete encryption schemes works in the spatial domain. The selective and joint compression and encryption schemes work in the transform domain as usually multimedia data is compressed before transmission. Because digital videos are generally compressed in DCT (discrete cosine transform) domain, almost all video encryption algorithms work in DCT domain. Due to the recent prevalence of wavelet compression technique and the adoption of wavelet transform in JPEG2000 standard in recent years image/video encryption algorithms working in wavelet domain have also attracted some attention.

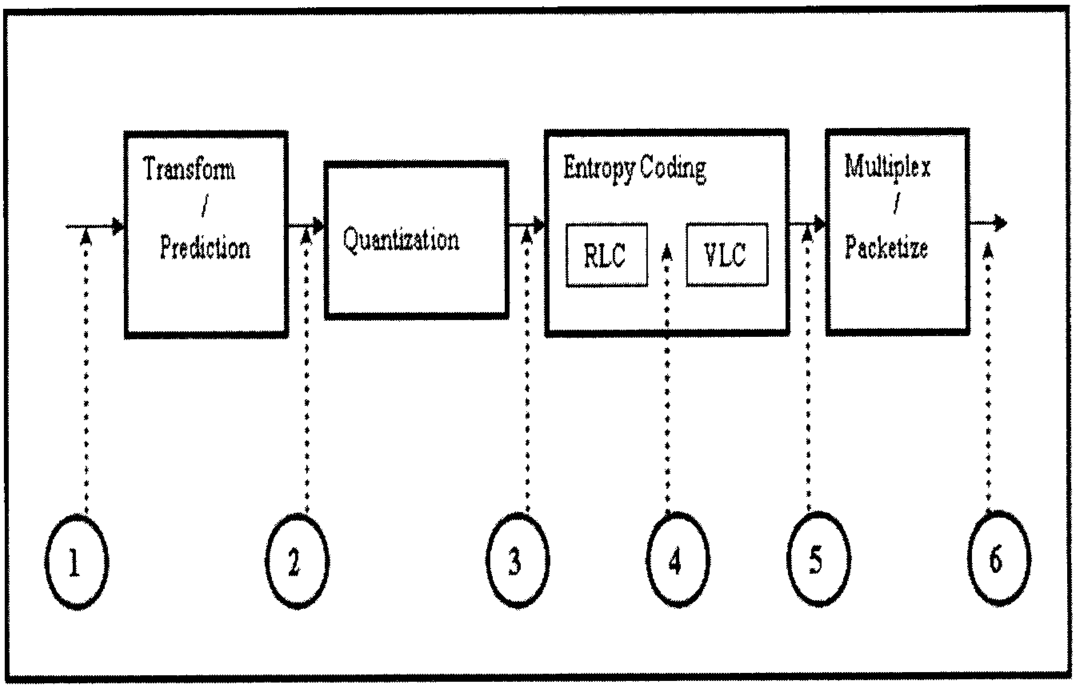


Fig. 2.1. Candidate domains for the various encryption algorithms [Wu & Mao, 2002].

Fig.2.1 highlights the candidate domains for the prevailing encryption schemes used by multimedia encryption community. As illustrated in Fig. 2.1, there are two straightforward locations to apply generic encryption (complete encryption schemes) to multimedia. The first possibility is to encrypt multimedia samples before any lossy or lossless compression i.e. stage #l in Fig. 2.1. Since the stage # 1 emphasizes use of encryption on uncompressed data, it defeats the use of the source coder as the encryption changes the statistical and structural characteristics of the original multimedia source significantly, resulting in much reduced compressibility of the data. However in multimedia encryption community, still this kind of approach is used for the encryption of the digital images as medical images and other files which are stored in the original format without compression can benefit from this type of application and protects data on the storage device from illegal viewing.

The second possible location for the generic ciphers (complete encryption schemes) is to encrypt the encoded bit stream after lossy or lossless compression, i.e., stage # 5 and stage # 6. This approach makes full use of compressibility of multimedia data and allows very high level of security.

In the case of selective and joint compression-encryption schemes, both compression and encryption is carried out at the same time. The candidate domains for the selective encryption schemes used in the community are stage #3 and stage#5 in Fig.2.1, under the assumptions and requirements considered by various researchers. The candidate domains for joint compression-encryption schemes are stage #2 and stage#4 in Fig. 2.1. Although many efforts have been devoted to better solutions for image and video encryption, the current security analyses of many schemes are not sufficient, especially on the security against known/chosen-plaintext attack. What's worse, many selective and joint compression- encryption schemes are insecure against ciphertext-only attack, due to the visible information leaking from unencrypted data.

In the following section, a comprehensive survey of the various non-chaos based multimedia encryption schemes for digital images, digital videos and digital audio, will be carried out, followed by prevailing chaos based encryption schemes for multimedia.

2.2 Non Chaos Based Multimedia Encryption Schemes

In this section, a comprehensive survey of the various prevailing non-chaos based multimedia encryption schemes is carried out. The schemes are classified into the three main types according to the domains specified in Fig. 2.1, where they are applied.

2.2.1 Complete Encryption Scheme

Complete encryption scheme as its name implies, encrypts the whole data stream. In the complete encryption schemes, usually the ciphers used are the standard block ciphers. Complete encryption schemes applied after or before lossy or lossless compression has higher computational load especially for multimedia data and some of the researchers have called these types of schemes as naïve [Qiao & Nahrstedt, 1997]. However, the complete encryption schemes provide very high level of security. In this section, a brief description of four significant complete encryption schemes is presented.

- An image encryption scheme is proposed in [Dang & Chau, 2000] that combines Discrete Wavelet Transform (DWT) and DES, where the bandwidth and redundancy is first lessened by compressing the image by Discrete Wavelet Transform (DWT) and then the pay load of ATM packets, carrying the compressed image, is encrypted by DES.

- FEA-M (Fast Encryption Algorithm for Multimedia) is proposed in [Yi et al., 2001]. FEA-M is based on Boolean matrix theory, where the size of input plaintext, ciphertext and secret key is equal to 64×64. However FEA-M has been proved insecure by Youssef et al. in [Youssef & Tavares, 2003]. The computational load for cryptanalysis is very low.

- An image encryption scheme is introduced in [Hou & Wang, 2003], which combines Discrete Wavelet Transform (DWT), Embedded Zero-tree Wavelet Algorithm (EZW), and AES. The proposed scheme is similar to the scheme presented in [Dang & Chau, 2005]. However in [Hou & Wang, 2003], image is first compressed through DWT and EZW and then output stream is encrypted with AES, instead of DES.

- An image encryption algorithm based on the closed form of an infinite series is introduced in [Balasubramanian, 2005]. The proposed scheme works before the compression process. The proposed scheme uses three keys κ_1, κ_2 and κ_3, where κ_1 is the main key and κ_2 and κ_3 are dependent on the particular image, equal to width and height of the image respectively. The proposed scheme has the following shortcomings: 1) if κ_1 is not significantly large, then image cannot be correctly restored. But if κ_1 is large, then the size of ciphered image will be very large; 2) required number of computations for encryption of an image is directly proportional to the size of the κ_1.

2.2.2 Selective Encryption Schemes

Selective encryption, as its name implies, encrypts the multimedia data selectively. Selective encryption schemes are those schemes that only encrypts portion of the multimedia data that carries the rich content. Since selective encryption encrypts only significant information and leaves insignificant information un-encrypted, it is regarded to provide a better tradeoff between the encryption load and the security level. Hence, since mid 1990s, various schemes have been proposed using selective approach for the encryption of the multimedia data. In [Tosun, & Feng, 2001], it was pointed out that selective encryption is also useful for realizing error-tolerating property in wireless video transmission. In addition, some other properties such as scalability [Kunkelmann & Reinema, 1997] are also realized via selective encryption schemes. In the following section, some of the selective encryption schemes are briefly explained.

- A partial encryption scheme for DCT based compression algorithms (JPEG, MPEG etc.) is proposed in [Kunkelmann & Reinema, 1997], where only higher order DCT coefficients are encrypted. The number of DCT coefficients to be encrypted is scalable, depending on the constraints of security and time. However the proposed scheme decreases the compression ratio.

- A generic video encryption algorithm (VEA) is proposed to encrypt video stream in [Qiao & Nahrstedt, 1997]. It consists of the following steps: divide each 128-byte piece of plain video into two 64-byte list: an Odd List and an Even List. Even list is XORed by the Odd list and Odd list is encrypted by a block cipher like DES, AES etc. The VEA can reduce the encryption load by 50%.

- Three more encryption algorithms- another VEA (Video Encryption Algorithm), RVEA (real-time VEA) and MVEA (Modified-VEA) [Shi & Bhargava, 1998a] [Shi et al., 1999] [Shi & Bhargava, 1998b] are proposed between 1998-1999. The short description of each one is given below:

    - In VEA only sign bits of DCT coefficients are encrypted

    - In MVEA uses a secret key randomly changing the sign bits of encoded differential values of DC coefficients of I pictures and the sign bits of encoded differential values of motion vectors of B and P pictures.

    - RVEA is a combined version of VEA and MVEA where the XOR operations are replaced by a traditional block cipher. For each macroblock at most 64 sign bits are encrypted with the order from low frequency to high frequency.

However, It is pointed out in [Li et al., 2004] that MVEA is weaker than VEA. VEA and RVEA are shown to be insecure in [Wu & Kuo, 2005b] and the security offered by the encryption of only sign bits is not sufficient

- Encryption schemes for images and videos compressed by quadtree and SPIHT compression algorithms are presented in [Cheng & Li, 2000]. However the proposed scheme is shown to be insecure in [Said, 2005].

- A multilayered video encryption scheme is proposed in [Tosun & Feng, 2000] based on VEA proposed by Qiao et al. in [Qiao & Nahrstedt, 1997] to support error tolerability property. The proposed scheme partitions the stream into three layers based on a break point consisting of two numbers (a, b). The resulting layers of video are called base layer, middle layer and enhancement layer and only base and middle layers are encrypted by VEA [Qiao & Nahrstedt, 1997].

- Various approaches for partial encryption of the MPEG-I based video are investigated in [Alattar & Al-Regib, 1999]. It is pointed out that: encrypting only the data in the Intra-coded frames does not provide adequate level of security [Spanos & Maples, 1996]. In 2nd method, Intra-coded macroblocks (I-MB's) in all frames are encrypted. However the motion content of the video sequence is still apparent. In 3rd method, headers of all predicted macroblocks (MBs) in addition to all I-MB's are encrypted. Comparatively the 3rd method has higher security.

- An encryption scheme for MP3 music is introduced in [Thorwirth, 2000] where the information associated with the most critical frequency bands is encrypted. In the proposed scheme low quality audio sample can still be heard but full quality music sample will be available only on decryption with the correct key.

- An extension of [Qiao & Nahrstedt, 1997] VEA is proposed in [Tosun, & Feng 2001] to additionally support the error-tolerability property. In extended VEA, error propagation is avoided by encrypting only one quarter of the data by cipher such as DES, AES etc.

- Encryption scheme for speech compressed by G.729 codec is proposed in [Servetti & Martin, 2002] [Servetti & Martin, 2002]. The proposed scheme can reduce the encryption load to 55 – 70 %.

- TSP (Take, Skip, Permute), is proposed in [Shieh, 2003] for the encryption of MPEG based video that operates on the output of Huffman encoding. TSP is designed on the assumption that since significant autocorrelation score comes every 8 to 32 bits in the Huffman entropy encoded stream, it takes 1 to 3 bits randomly at the beginning of the stream and selectively skips 8 to 32 bits in order to take the next 1 to 3 bits. The process is repeated until the end of the encoded stream. The selected bits are permuted and then restored to the positions where the bits had been taken from.

- In [Seo et al., 2003] encryption of sign bits of partial DWT coefficients is suggested. However this scheme is not secure as encryption of only sign bits do not provide sufficient security and is even insecure against the ciphertext only attack.

- An encryption scheme for MPEG audio is proposed in [Servetti et al., 2003]. The proposed scheme consists of two main steps: 1) directing the decoder to skip decoding some regions; 2) selectively encrypting region '0' and region '1'. The aim is to provide listeners with sample-quality audio material that can be upgraded to full-quality by simply acquiring key and decrypting selected bits. Selected bits are only 1-10% of the already available data.

- The scheme proposed in [Liu & Li, 2004] emphasizes the importance of the motion vectors in the encrypting the video. Based on analysis and experiments the importance of the data in motion vectors is illustrated. In [Liu & Li, 2004] a new motion vector encryption algorithm called MVEA is proposed that consists of two main steps: 1) concealing statics; 2) spatial distancing.

- A secure color image coder was proposed in [Martin et al., 2005], employing the C-SPIHT compression algorithm and a stream cipher, where the confidentiality is achieved by encrypting only the significant bits of individual wavelet coefficients. Unencrypted bits are difficult to interpret due to its dependency on encrypted significant bits (Error propagation). The user may control the confidentiality versus processing overhead by choosing a parameter, K, at the time of encrypting.

- Different options in encrypting the JPEG images are analyzed in [Vancea et al., 2005]. It is pointed out that the best way to encrypt is permutation of the stream combined with XORing with a key stream.

From the above survey, it is seen that most of the selective multimedia encryption schemes are broken. Hence, it seems difficult to achieve high security using the selective encryption approach. It is only suitable to realize perceptual encryption, i.e., to degrade the visible quality of the encrypted plain-image/video, rather than to provide cryptographically strong security.

2.2.3 Joint Compression and Encryption Schemes

Though it is difficult to differentiate between the selective and joint compression-encryption schemes, in [Wu & Mao, 2002] an attempt has been made to differentiate between the two. According to [Wu & Mao, 2002], the candidate domains for joint compression-encryption schemes are stage # 2 and stage # 4 in Fig. 2.1.

- An MPEG video encryption algorithm is proposed in [Tang, 1996], where DCT data are read randomly instead of zigzag order during the MPEG compression process. However the problems related with the proposed scheme [Tang, 1996], are identified in [Qiao & Tam, 1997]. It is shown that 1) the encryption method causes a significant increase in size of the MPEG video stream, defeating the purpose of compression, and 2) the encryption method can not withstand the known-plaintext attack.

- An encryption scheme is proposed in [Wu & Kuo, 2001a], where multiple Huffman Tables (MHT) are employed and a random index is used as the secret key to select a Huffman table for each codeword. This scheme cannot resist the chosen-plaintext attack [Li et al., 2004].

- An encryption algorithm is proposed in [Kankanhalli & Guan, 2002], that consists of the following two main steps: 1) shuffling the code words within each of the Huffman tables; 2) random flipping of the last bit of the code word.

- Another encryption scheme is proposed in [Wen et al., 2002], where the Huffman tables are left untouched and the secret entropy codec is achieved as follows: map each plain-codeword to an index representing its position in the Huffman table, encrypt the index, and then re-map the encrypted index back to generate cipher-codeword.

- An encryption scheme is proposed in [Zeng & Lei, 2003] for the wavelet compressed videos that consists of the following three operations: 1) selective bit scrambling; 2) block shuffling and 3) block rotation. However the proposed scheme can not withstand the chosen plaintext attack [Li et al., 2004].

- Three permutation only methods for image encryption are compared and a new method is proposed in [Lian & Wang, 2003]. They are: 1) CWW [Uehara, 2001] - where wavelet coefficients are permuted among the whole image; 2) CWF [Zeng & Lei, 2003] - where wavelet coefficients among the same frequency subband are permuted; 3) CWQ- [Lian & Wang, 2003] - where wavelet coefficients among the child nodes (having the same parent node) are permuted. However such permutation-only methods are not secure enough and the secret permutations can be reconstructed under the known/ chosen plaintext attack.

- An encryption scheme for MPEG video is proposed in [Cheong, 2005] that consists of the following steps: 1) Huffman tree mutation step – where on the basis of subpart K_1 of the secret key, a new Huffman table is created; 2) rotating the i^{th} byte in the encoded stream, by using the subpart K_2 of the secret key as seed for the random permutation generator; 3) XORing the rotated stream with random sequence, where subpart of K_3 is used as seed for random sequence generator.

It is seen that different approaches are used for encryption of multimedia using joint compression and encryption approach. In this case, encryption schemes are dependent on the compression schemes, hence, for different compression schemes, different encryption schemes have to be designed. However, the computational complexity of most proposed schemes is low compared to complete encryption schemes. Some of the proposed schemes are also reported to have been broken later. Hence further studies are still needed to validate the security level of joint compression and encryption schemes.

2.3 Chaos Based Multimedia Encryption Schemes

Due to the tight relationship between chaos and cryptography, it is being widely investigated as to how one can use chaotic maps to construct the new and efficient cryptosystems especially for the multimedia data since 1989. An avalanche of work since then demonstrated various multimedia encryption schemes in the literature. In the prevailing schemes, the chaotic maps are used in two basic ways: 1) use chaos as a source to generate pseudo-random bits with desired statistical properties to realize secret masking based encryption operations; 2) use 2-D or 3-D chaotic maps to permute the pixels of the digital images around in some secret way. The first way has been widely used to design chaotic stream ciphers, while the second is specially employed by chaos-based image encryption schemes for dynamically scrambling the image.

2.3.1 Complete Encryption Scheme

As already explained, in complete encryption schemes entire image is encrypted by the cryptosystems based on 1-D, 2-D, 3-D chaotic systems or combination thereof. Cryptosystems based on single chaotic systems are not regarded to be secure [Sobhy, & Shehata, 2001 (a)] and recently most of the chaotic cryptosystems are developed based on multi chaotic systems. 1-D chaotic systems are used as source to generate pseudo-random chaotic key stream with the desired statistic properties. The chaotic key stream is used to mask the pixel value of plaintext [Pareek et al. 2005]. These types of schemes are referred as substitution-only schemes. 2-D and 3-D chaotic maps are used to dynamically permute the pixel values of the digital image, called as permutation only schemes. However, the cryptosystems based on only substitution or only permutation have been shown to be insecure against the ciphertext, known/chosen-plaintext attacks.

In this section, a brief description of the various chaos based cryptosystem for digital images is presented. The schemes are further classified on the basis of the dimension of the employed chaotic systems i.e. 1-D, 2-D or 3-D.

2.3.1.1 Image Encryption Schemes based on 1D Chaotic Maps

In this section, various chaos based image encryption schemes will be briefly discussed. Though every chaotic cryptosystems can be used for the encryption of the digital images, the focus in this section will only be on those schemes that are specially designed for digital images.

- Several image encryption schemes have been proposed by Yen et al. in [Yen & Guo, 1999] [Guo & Yen, 1999] [Cheng & Gum, 2000] [Yen & Guo, 2002]. All of the schemes are based on the logistic map $f(x) = ux(1-x)$. The secret key includes initial condition $x(0)$ and control parameter u (for some schemes, the secret key includes some additional parameters). All of these schemes follow similar basic idea of using chaos to realize the encryption: 1) run the logistic map to generate the pseudo-random binary sequence; 2) use the generated chaotic binary sequence to pseudo-randomly control permutation and/or substitution for all pixels.

  - **BRIEA** (Bit Recirculation Image Encryption Algorithm) [Yen & Guo, 1999]: It is based on the chaotic binary pseudo random sequence that is used to shift each pixel to a new position. The proposed scheme was successfully cryptanalyzed in [Li & Zheng, 2002 (a)]. It is seen that it cannot resist the known/chosen - plaintext attacks, since a mask image that is equivalent to the secret key can be derived from a pair of plain image and cipher image.

  - **MLIE** (Mirror-Like Image Encryption) [Guo & Yen, 1999]: Based on a binary sequence generated from a chaotic system, an image is scrambled according to certain algorithm. This algorithm consists of 7 steps. Step-1 determines a 1-D chaotic system and its initial point $x(0)$ and sets $k = 0$. Step-2 generates the chaotic sequence from the chaotic system. Step-3 generates binary sequence from chaotic system. Steps-4, 5, 6, and 7 rearrange image pixels using swap function according to the binary sequence. But since the proposed scheme is only permutation based, it can not resist the known/ chosen plaintext attacks [Li et al., 2004].

- **CKBA** (Chaotic Key Based Algorithm) [Cheng & Guo, 2000]: Based on the chaotic pseudo random sequence, *key*1 or *key*2 are dynamically XORed (XNORed) with each pixel of the image. However the proposed scheme is insecure against the chosen/ known-plaintext attack and the security against the brute force attack is also less than the one claimed in CKBA [Li & Zheng, 2002 (b)].

- **TDCEA** (The 2D Circulation Encryption Algorithm) [Yen & Guo, 2002]: A new scheme is introduced based on permutation and value transformation mechanism. The proposed scheme is an enhanced version of BRIEA [Yen & Guo, 1999 (a)], and a bit shift operation controlled by chaotic pseudo random binary sequence are applied in four different directions on a bit-matrix M composed of 8 consecutive pixel values. Though the TDCEA is more secure than BRIEA, it is still not secure enough against the chosen plaintext attack.

Also the generated chaotic pseudo-random sequence is not balanced, i.e., the number of 0's is different from that of 1's, due to the non-uniform density function of the Logistic map. Therefore the Logistic map is not a good choice for encryption, so it is better to use other 1-D chaotic maps with uniform probability density function.

- An encryption scheme has been proposed in [Belkhouche & Qidwai, 2003] for binary images based on 1-D chaotic system. The proposed scheme is permutation only scheme. The following chaotic system is used in the proposed scheme:

$$x_{n+1} = \sin\left(\frac{a}{x_n}\right) \quad \text{with initial condition } x_{n=0} = x_0 \tag{2.1}$$

According to the pseudo-random sequence generated by the map in (2.1), pixel values are dynamically permuted. Since the proposed scheme is permutation only scheme, it can not resist the known / chosen-plaintext attacks [Li et al., 2004].

- Yet another image encryption scheme that operates on pixel values is proposed in [Pareek et al. 2006 (b)]. In the proposed scheme an external secret key of 80-bit and two chaotic logistic maps are employed. The initial conditions for the both logistic maps are derived using the external secret key by providing different weight-age to all its bits. Further, in the proposed encryption process, eight different types of operations are used to encrypt the pixels of an image, and which operation will be used for which pixel, is decided by the outcome of a logistic map. The secret key is modified after encrypting each block of sixteen pixels of the image. However the proposed scheme is not suitable for real time encryption because the number of iterations and operations employed to encrypt a pixel is very high. In the Chapter 3 of this dissertation, the proposed scheme is shown to be insecure against differential type of attack.

2.3.1.2 Image Encryption Schemes based on 2D Chaotic Maps

The idea of using 2-D chaotic maps to design permutation based image encryption was initially proposed in [Pichler & Scharinger, 1996] [Scharinger, 1997] [Scharinger, 1998] and later systemized in [Fridrich, 1997 (a) &, 1998 (b)].

- The proposed scheme in [Fridrich, 1997 (a) &, 1998 (b)] works in the following way for an image of the size M × N.
  - Define a discretized and invertible 2-D chaotic map on an M × N lattice, where the discretized parameters serve as the secret key;

- Iterate the discretized 2-D chaotic maps on the plain-image to permute all pixels;

- Use a substitution algorithm (cipher) so as to modify the values of all pixels to flatten the histogram of the image.

- Repeat the permutation and the substitution for K rounds to generate the cipher-image.

The map used in the proposed scheme in [Fridrich, 1997] [Fridrich, 1998] is the Baker map which is given by (2.2). It is also possible to use the Cat map and Standard map [Fridrich, 1997] [Fridrich, 1998].

$$B(x,y) = \begin{cases} (2x, y/2) & 0 \le x < 1/2 \\ (2x-1, y/2+1/2), & 1/2 \le x < 1 \end{cases} \tag{2.2}$$

However, in [Salleh et al., 2002 (a)], it is pointed out that there exists weak keys in the schemes proposed in [Fridrich, 1997] [Fridrich, 1998]. To overcome this defect a modified scheme is proposed in [Salleh et al., 2002 (a)] that consists of the following steps: 1) Binding of password to image; 2) Permuting the image by Baker's map; 3) Nonlinear feedback substitution; 4) Shifting pixels in rows. The fourth step avoids the recurrent short period of the discretized chaotic permutation.

- Zhang et al. introduce a map defined by a new matrix called T- matrix [Zhang et al., 2004]. For $n \ge 2$, the T-matrix is generated as:

$$T(1,j) = 1,\ T(i,1) = 1, \quad i,j = 1,2,\ldots,n$$

$$T(i,j) = T(i-1,j-1) + T(i-1,j), \quad i,j = 2,3,\ldots n$$

That is

$$T_{n\times n}=\begin{bmatrix}1 & 1 & 1 & \cdots & 1 & 1 & 1\\ 1 & 2 & 2 & \cdots & 2 & 2 & 2\\ 1 & 3 & 4 & \cdots & 4 & 4 & 4\\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots\\ 1 & a_1 & a_2 & \cdots & a_{n-3} & a_{n-2} & a_{n-1}\\ 1 & 1+a_1 & a_1+a_2 & \cdots & \cdots & a_{n-3}+a_{n-2} & a_{n-2}+a_{n-1}\end{bmatrix} \quad (2.3)$$

The periods of a Cat map and T-matrix are compared in [Zhang et al., 2004] and it is found that the period of the T-matrix is higher than that of Cat map when $N=2^{\alpha}$. However, till now only limited results on a subset of all chaotic possible permutations have been reported. Therefore further theoretical research is needed to extensively study the security of the image encryption schemes based on it. However some work reported are not encouraging [Qi et al., 2000] [Li et al., 2004].

- An image encryption algorithm based on the line map is formulated in [Feng et al., 2006]. The image is permuted based on a line map. In the proposed scheme, the secret key determines the number of iteration of the left and right line maps. For the deciphering process, the inverse of the line map is applied with the same number of iterations. To bring the substitution property, pixel value is modified by XORing with index values i and j, representing the row and column number respectively.

2.3.1.3 Image Encryption Schemes based on 3D Chaotic Maps

- An image encryption scheme based on a more complex 3-D chaotic cat map is proposed in [Chen et al., 2004]. The 3-D cat map is formulated by extending the

2-D cat map and is considered to be more secure due to higher Lyapunov exponents. The designed scheme consists of the following steps: 1) Key generation; 2) Converting the 2-D image into 3-D; 3) Permute the 3-D image by 3-D cat map; 4) Diffusion process by using chaotic logistic map, along with ciphertext feedback; 5) Transform back the 3-D image back into 2-D form. The 3-D cat map used in step 3 is given below:

$$A=\begin{bmatrix} 1+a_x a_z b_y & a_z & a_y + a_x a_z + a_x a_y a_z b_y \\ b_y + a_x b_y + a_x a_z b_x b_z & a_z b_z + 1 & a_y a_z + a_x a_y a_z b_y b_z + a_x a_y b_y + a_x \\ a_x b_x b_y + b_y & b_x & a_x a_y b_x b_y + a_x b_x + a_y b_y + 1 \end{bmatrix} \tag{2.4}$$

where a_x, b_x, a_y, b_y, a_z and b_z are all positive integers and serves as control parameters for the 3-D generalized cat map. The proposed scheme consists of both the permutation and substitution mechanism but the proposed system has higher computational load.

- [Fei et al., 2005] proposed an image encryption scheme based on multiple 3-D chaotic systems. The following three chaotic systems are employed in the proposed algorithm, which are selected dynamically.

$$\begin{array}{ccc} x' = a(y-x) & x' = a(y-x) & x' = a(y-x) \\ y' = cx - xz - y & y' = (c-a)x - xz + cy & y' = -xz - cy \\ z' = xy - bz & z' = xy - bz & z' = xy - bz \\ \text{(a)} & \text{(b)} & \text{(c)} \end{array} \tag{2.5}$$

Here (a) is the Lorenz system, (b) is the Chen system and (c) is the LÜ system. The proposed algorithm however has reduced key space size which is explained in chapter 3 of this thesis. The proposed scheme fights against the known plaintext attack by introducing a ciphertext feedback. Also the encryption speed of the proposed scheme is very low due to difficulty in realizing chaotic systems employed.

2.3.2 Joint Compression and Encryption Schemes

- A chaotic video encryption scheme is proposed in [Chiaraluce et al., 2002] for encrypting the following selective data of an H.263+ encoded video: most sign bits of DC coefficients, the AC coefficients of I-macroblocks, sign bits of AC coefficients of P-macroblocks, and sign bits of motion vectors. The proposed scheme is a stream cipher based on three different chaotic maps: the skew tent map, the skew saw tooth map, and the discretized Logistic map. The outputs of the first two chaotic maps are added, and then the addition is scaled to be an integer between 0 and 255. Each scaled integer is used as the initial condition of the third map to generate a 64-byte key stream to mask the plaintext with XOR operation. To further enhance the security against known/chosen-plaintext attacks, it was suggested to change the key every 30 frames.

- An image encryption scheme is proposed in [Lian et al., 2004 (a)], based on JPEG source encoding. The proposed scheme consists of the following steps: 1) DCT blocks are diffused using Pseudo-Random Space Filling Curves (PR-SFCs); 2) DCT coefficients are diffused according to different frequency bands; 3) signs of DCT coefficients are encrypted by a chaotic stream cipher.

- An MPEG based video encryption algorithm is proposed in [Lian et al., 2004 (b)] which consists of the further sub-algorithms:

  - CREA (Chaotic Run-Length Encryption Algorithm) – it is based on 2-D one-way coupled map lattice (OCOML) and encrypts run length codes by 2-D chaotic sequence.

- 2) SECSC (Security Enhanced Chaotic Stream Cipher), – it is based on the logistic map and encrypts the signs of motion vectors
- 3) CKD (Chaotic Key Distributor) – it is based on the logistic map and distributes keys among GOPs (Group of Pictures), slices and frames.

The proposed scheme is extended for MPEG2 in [Lian et al., 2004 (c)] and shown that the proposed scheme is more robust to transmission errors than VEA [Qiao & Nahrstedt, 1997].

- An encryption scheme based on JPEG2000 codec is proposed in [Lian et al., 2004 (d)], that permutes different number of bit-planes in each code block, encrypts different number of coefficients' signs, and permutes code blocks in different frequency bands. Thus, the images or videos can be degraded to different degrees under the control of quality factor.

- A chaos-based adaptive arithmetic coding-encryption technique is designed in [Bose & Pathak, 2006]. For compression, it relies on a standard zeroth-order adaptive arithmetic coder, while for encryption, the arithmetic coder's statistical model is turned into variable nature using the bit stream generated by CCS PRBG, developed in [Li et al., 2004]. The designed model dynamically reorders the frequency of the input symbols according to the CCS PRBG and depends on all the text that has been coded since the initialization of the model.

2.3.3 Generic Chaotic Encryption Schemes

- A Multi Layer Chaotic Encryption (MCE) algorithm for the packet switched networks is proposed in [Beritelli et al., 2000]. The structure of the designed cipher is multilayer, where initial conditions and control parameters of the next layer chaotic systems depend on the output of the previous layer chaotic systems. However initial condition and control parameters of the first layer chaotic systems depend on the external secret key. The proposed scheme has been shown to be apt for the transport layer security (TLS) protocol. However the proposed scheme can be shown to be insecure against the known plaintext attack, though the secret key extraction is difficult but the secret chaotic key stream (equivalent key) can be extracted easily.

- A new stream cipher is designed in [Klomkarn et al., 2004] based on multi-chaotic functions to safeguard against the attack proposed in [Sobhy & Shehata, 2001 (a)]. The proposed stream cipher is based on the following chaotic maps – Logistic map, Chebeshev map, Iterative Map with Infinite Collapses (IMIC), Gauss map and Cat map. In the proposed cipher, a combination of systems is selected by the cryptographic key from a set of already defined combinations. Through selected combination of chaotic systems, another combination of systems and their running time is selected, which is eventually used for generating the random key stream.

- A chaotic block cipher is proposed in [Pareek et al. 2005 (a)] based on multiple one-dimensional chaotic maps as the cryptosystems based on single chaotic maps have been shown to be insecure [Sobhy & Shehata, 2001 (a)]. In the proposed block cipher, following map were used – logistic, tent, sine and cubic. However the proposed cipher has been shown to be insecure against the known plaintext attack and has a reduced key space size, as shown in chapter 3 of this thesis. Also the encryption speed of the proposed cipher is very slow due the fact that the number of iterations used by the chaotic systems depends on a dynamically selected ASCII value of the character in the secret key.

2.4 Summary

From the study of the chaos based multimedia encryption schemes (especially images), one can learn lessons on how to design fast and secure multimedia encryption schemes with or without using chaos. Although some experiences and lessons are commonplace in modern cryptography, for some reasons they have not been widely accepted by the multimedia encryption community. In what follows, we summarize these lessons.

- *Encrypting only sign bits of DCT coefficients does not provide sufficient security.*

  In case of partial image or video encryption schemes, it has been examined that sign bits of DCT coefficients are either encrypted with a more secure block cipher e.g. AES etc or their values are changed randomly by a stream cipher. However, in either case, the said method of encryption is not secure against ciphertext-only-attack.

- *Multimedia encryption schemes based on masking only mechanism are insecure against known plaintext attack.*

  In case of chaos based encryption schemes, it is seen that pixel values of images are XORed with secret keys dynamically with the help of one dimensional chaotic maps. However under known-plaintext attack, a mask image that is equivalent to the secret key can be obtained to fully decrypt cipher images encrypted with the same secret key, if the size of the ciphered images is less than or equal to size of the mask image.

- *Multimedia encryption schemes based on permutation only mechanism are insecure against known/chosen plaintext attacks.*

  For encryption of multimedia, secret permutation (scrambling) are often used by multimedia encryption community. However, in our opinion, it is not a prerequisite for a secure multimedia encryption schemes. In most of the schemes, images are scrambled dynamically by higher order dimensional chaotic maps that lead to their higher computational load. Permutations-only schemes are also unable to change plain-histograms, which contain some useful information about the plain images that can be used in the attack. For example, images of human faces usually have narrower histograms than photons of natural scenes. In addition, the secret permutations can be broken under the known/chosen plaintext attacks.

- *For enhancing the security against known/chosen-plaintext attacks, ciphertext feedback is very useful.*

  In recent image encryption schemes in [Chen et al., 2004] [Fei et al., 2005], ciphertext feedback is employed to resist known/chosen plaintext attacks. Also in the appendix A of this dissertation, it is shown that CKBA can be made more secure by employing ciphertext feedback.

- *Avalanche property at the level of image can be achieved by employing ciphertext feedback.*

  Most of image encryption schemes do not fulfill the avalanche property at the level of image (a change in one pixel should be diffused over rest of the pixels within the image) [Pareek et al., 2006] [Fei et al., 2005]. In order to create a diffusion mechanism, ciphertext feedback along with the number of rounds greater than or equal to 2 should be used.

- *Most chaotic encryption schemes lack rigor as they appear to be intuitively designed.*

  It has been observed that the design of most chaotic schemes is motivated by the apparent random properties of chaos. The schemes are intuitive and ad hoc. Besides, the detailed cryptanalysis of most of chaotic schemes is inadequate. This is due to the fact that cryptanalysis tools are still under development in the field of chaotic cryptography. However, we believe, the conventional cryptanalysis tools can also be used sometimes to prove the security against certain attacks.

- *For higher security, most of the chaos-based image encryptions schemes rely on higher number of time consuming iterations of chaotic maps.*

It is observed that for higher security, chaos-based schemes rely on higher number of iterations. For example in Pareek et al.'s image encryption scheme [Pareek et al., 2006], number of iterations used for encryption of a pixel are dependent on K_{10}, whose value can be between 1 and 255. If the value of K_{10} is low, then the security of the proposed scheme is very low. In addition to higher number of iterations, some of the schemes use higher dimensional chaotic maps for encryption of images. Such higher dimensional maps are not convenient for implementation. For example in case of Fei et al.'s image encryption scheme [Fei et al., 20005], 3-D chaotic maps are used and number of iterations used for a pixel are between 5 and 29. The employed systems are iterated by 4th order Runge-Kutta method that includes multiplications and additions at each step. We discovered that the throughput of the proposed scheme is very low and does not fulfill the requirement of real time encryption of digital images. In our opinion, simplest 1-D chaotic maps should be employed for designing encryption algorithms, as they are convenient for implementation.

- *Chaotic ciphers are considered simpler to implement.*

It is observed that most chaos based image encryption schemes claim that the throughput of their ciphers is higher than that of traditional ciphers e.g. DES, AES etc. and that is because chaotic maps are considered simpler to implement compared to traditional ciphers.

- *S-box equivalent security not reported.*

It is also observed that in the security analysis of chaos based encryption schemes, no one has reported S-box equivalent security in their ciphers. S-box is a nonlinear transformation, applied on a bundle of bits. The bundle of bits many be 3, 4, or 8 etc, depending on the design of cipher. In case of AES S-box, bundle size is 8-bit. S-box is designed according to some basic criteria, given as:

- ***Non-Linearity.***
    a) **Correlation.** The maximum input-output correlation amplitude must be as small as possible.
    b) **Difference propagation probability.** The maximum difference propagation probability must be as small as possible.
- **Algebraic complexity.** The algebraic expression of S-box has to be complex.

For AES S-box, maximum correlation amplitude is 2^{-3}, and the maximum difference propagation probability can be as low as 2^{-6}. However, in most of chaos based schemes, no report has been made as to what the maximum correlation amplitude of their cryptosystem is, or what the maximum difference propagation probability is. In cryptanalysis of some chaos based scheme in chapter 3 of this thesis, it is shown that chaos based schemes are unable to distribute the differentials effectively. It is also evident that most of the schemes are based on *masking only* mechanism, which has no affect on the difference propagation probability.

From the above survey and discussion, it is evident that chaos theory has been widely investigated for the encryption of digital images and a number of encryption algorithms have been proposed recently. However, most of the chaos-based image encryption schemes have either been broken, or not suitable for the real-time encryption of digital images. Hence we are in need of a suitable image cipher, which could fulfill the requirements of real-time secure encryption digital images.

3

CHAPTER

Cryptanalysis of Some Chaos Based Encryption Schemes

3.1 Introduction

In this chapter, the cryptanalysis of a few recent chaotic ciphers is presented. It is observed that in the last decade, many chaotic ciphers that have been proposed have proved to be insecure. The cryptanalysis of these representative chaotic ciphers is undertaken to gain better understanding of the weaknesses in the hope that this would enable a better design of the ciphers.

3.2 Cryptanalysis

Cryptanalysis is the science that evaluates the promises of cryptography. When we make a purchase over the Internet, we would like our transaction to be "secure". We wish that attackers would not be able to tap personal information or change the results of electronic voting without being detected or make fraudulent calls on our account, etc. Cryptanalysis focuses on evaluating the strength of cryptographic algorithms.

In practice, cryptography and cryptanalysis are viewed as two sides of the same coin: in order to create secure cryptography, one has to design against possible cryptanalysis as it provides the certificate of assurance for the security of the cryptographic algorithms. From a cryptographical point of view, a secure cipher should have the capability to weather all kinds of attacks. In most cases, the designed cipher should be secure against the brute-force attack (i.e., the attack of exhaustively searching all possible keys), and the following four typical attacks:

- **Ciphertext-only attack:** An attempt to decode the message P (plaintext) using the corresponding ciphertext C.
- **Known-plaintext attack:** An attempt to decode the current plaintext using the current ciphertext and previous samples of plaintext and their corresponding ciphertext. This attack is more likely to succeed than the ciphertext-only attack. In this type of attack the unauthorized receiver knows what the plaintext was for some previous messages, however he can not choose those plaintext samples. Therefore the samples he has may or may not enable him to decode the current message and recover the secret key.
- **Chosen-plaintext attack:** This attack is similar to the known-plaintext attack, only this time the unauthorized receiver can choose the plaintext and observe the corresponding ciphertext. This attack gives the unauthorized receiver a better chance to break the encryption scheme since he can choose plaintext samples that will reveal the secret key as much as possible.
- **Chosen-ciphertext attack:** In this attack, the cryptanalyst chooses a ciphertext and causes it to be decrypted with an unknown key.

Most of the cryptanalytic attacks search for an identifiable non-random property in the structure of the cipher, or for the correlation among the available plaintexts and their corresponding ciphertexts. For example differential cryptanalysis (chosen-plaintext attack) studies the evolution of differences during encryption of two plaintexts (under the same encryption key). In particular, the XOR-difference during the different rounds of encryption is analyzed: starting with the difference of the two plaintexts, continuing through differences of intermediate values, and ending with the difference of the ciphertexts. Linear cryptanalysis studies the correlation of parities of data bits (plaintext and ciphertext bits) during the encryption process.

3.3 Cryptanalysis of Some Representative Chaotic Ciphers

We now undertake the cryptanalysis of some recent and representative ciphers based on chaos. There are 3 chaotic ciphers that we analyze in subsequent sections: Pareek et al's Chaotic Block Cipher [Pareek et al, 2005], Fei et al's Chaotic Image Cipher [Fei et al., 2005] and Pareek et al's Image Encryption Scheme [Pareek ae al, 2006].

3.3.1 Pareek et al.'s Chaotic Block Cipher

Pareek et al. proposed a time variant chaotic block cipher in [Pareek et al., 2005], based on the multiple chaotic maps. It is done because the cryptosystems based on the single chaotic systems are considered to be insecure. Another distinctive feature of the proposed cipher is that it does not explicitly use the system parameter or initial condition of the chaotic map as a secret key as other chaotic cryptosystems do. Before the present cryptosystem, Pareek et al. had proposed another cipher based on one dimensional chaotic logistic map in [Pareek et al., 2003]. They claim that the current cryptosystem is faster and more secure than their previous cryptosystem [Pareek et al., 2003] and even other chaotic cryptosystems proposed by other researchers [Baptista, 1998] [Wong et al., 2000] [Wong, 2002]. However, it is found that the proposed cryptosystem has a small key space size and is insecure against the proposed differential and known plaintext attacks. In what follows the said cryptosystem is systematically presented and analyzed.

3.3.1.1 STEP A

For the encryption/decryption, plaintext/ciphertext is divided into blocks of 8-bits and then grouped together in such a way that the group length is variable i.e. number of blocks in each group is different. Plaintext and ciphertext of '*n*' blocks is represented as:

$$P = \underbrace{P_1 P_2}_{G_1} \underbrace{P_3}_{G_2} \underbrace{P_4 \cdots\cdots}_{G_3} P_n \text{ (Plain text)}, \tag{3.1}$$

$$C = \underbrace{C_1 C_2}_{G_1} \underbrace{C_3}_{G_2} \underbrace{C_4 \cdots\cdots}_{G_3} C_n \text{ (Cipher text)}, \tag{3.2}$$

Secret key of 128-bits is divided into blocks of 8-bits named as session keys as:

$$K = K_1 K_2 K_3 K_4 \cdots\cdots K_{16} \tag{3.3}$$

The secret key is expressed in hexadecimal mode, so a 128-bits key will consist of total 32 alphanumeric characters (out of 0 to 9 and A to F) and each group of two alphanumeric characters represents the session key.

3.3.1.2 STEP B

The cipher uses following multiple one-dimensional chaotic maps – logistic map, tent map, sine map and cubic map and for convenience, each chaotic map is identified by an integer index (map number N) i.e. logistic map by an integer 0, tent map by 1, sine map by 2 and cubic map by 3. Table 3.1 represents the map number (N), governing equation and system parameter value in the chaotic range for each map used in the propose chaotic block cipher.

3.3.1.3 STEP C

Initially, same initial condition (*IC*) is chosen for each chaotic map which is generated from the session keys as:

$$R = \sum_{i=1}^{16} \left(K_i \big/ 256 \right) \tag{3.4}$$

$$IC = R - \lfloor R \rfloor \tag{3.5}$$

Here K_i is the decimal equivalent of the i^{th} session key and $\lfloor \cdot \rfloor$ is the floor function.

3.3.1.4 STEP D

A dynamic table DT1 is created, having number of rows equal to the total number of maps (four) used in the algorithm and two columns – one for map number (N) and another for initial condition for the chaotic map N. It is called dynamic because, with the encryption/decryption of each block of plaintext/ciphertext, the second column containing the initial condition for the chaotic map is updated and the updating depends on the encryption/decryption of the previous block of plaintext/ciphertext (explained in Step F). Initially, all the entries in the second column of DT1 are exactly same and equal to the initial condition value generated in the step C.

3.3.1.5 STEP E

Another dynamic table DT2 is also created, having number of rows equal to the total number of session keys and three columns – first column for number of blocks (B) of plaintext/ciphertext, second for the map number (N) and the last one for number of iterations (IT). The dynamic table DT2 is filled with the help of linear congruential random number generator (LCG), which generates a sequence of random numbers with the following recurrence relation:

$$Y_{n+1} = (aY_n + c) \bmod m, \tag{3.6}$$

where a, c and m, respectively, are multiplier, increment and modulus and are taken to be 5, 1 and 16 throughout the algorithm. The seed Y_0 for the LCG is a secret key dependent integer and is given by $Y_0 = \lfloor IC \times 10^2 \rfloor$, here IC is initial condition generated in Step C. The values of B, N and IT described above are generated from Y_{n+1} as

$$B = Y_{n+1}, \tag{3.7}$$

$$N = Y_{n+1} \bmod 4, \tag{3.8}$$

$$IT = decimail \quad equivalent \quad of \;\; the\, ((Y_{n+1} \bmod 16) + 1) th \; session \; key \tag{3.9}$$

3.3.1.6 STEP F

During encryption/decryption of plaintext/ciphertext, the dynamic tables (DT1 and DT2) are read as follows: A group of B blocks (no. of blocks) of plaintext/ciphertext is encrypted/ decrypted by iterating map N with initial condition IC and number of iterations IT. The values for B, N and IT are read from DT2 and the corresponding initial condition IC for map N is read from DT1. The new value of X_{new}, after IT iterations of map N, is used for encryption/decryption of plaintext/ciphertext as follows:

$$C_i = \left(P_i + \lfloor X_{new} \times 10^5 \rfloor\right) \bmod 256, \tag{3.10}$$

$$P_i = \left(C_i + 256 - \left(\left\lfloor X_{new} \times 10^5 \right\rfloor \bmod 256\right)\right) \bmod 256 \tag{3.11}$$

where P_i and C_i are, respectively, the ASCII values corresponding to the ith block of plaintext and ciphertext.

After encryption/decryption of each block of plaintext/ciphertext, the value of initial condition IC for map N in the dynamic table DT1, used in the recent encryption/decryption of plaintext/ciphertext, is replaced with the new value of X, X_{new}. In this way, the dynamic table DT1 is regularly updated after encryption/decryption of each block of plaintext/ciphertext.

Table 3.1. Chaotic systems used in Pareek et al.'s block cipher: Map number, governing equations and system parameters values in chaotic range for the one-dimensional maps used in the Pareek et al.'s cryptographic algorithm

Chaotic map	Map number (N)	Governing equation	System parameter value (λ) in chaotic range
Logistic	0	$X_{n+1} = \lambda X_n(1 - X_n)$	$\lambda = 3.99$
Tent	1	$X_{n+1} = \begin{cases} \lambda X_n & \text{if } X_n > 0.5 \\ \lambda(1 - X_n) & \text{if } X_n \le 0.5 \end{cases}$	$\lambda = 1.97$
Sine	2	$X_{n+1} = \lambda \sin(\pi X_n)$	$\lambda = 0.99$
Cubic	3	$X_{n+1} = \lambda X_n(1 - X_n^2)$	$\lambda = 2.59$

3.3.1.7 STEP G

When dynamic table DT2 is completely exhausted i.e. encryption/decryption of total S (sum of all the values in column one of DT2) blocks is achieved, the DT2 is filled again with the help of LCG as explained in Step E. This time the value of X_{new} obtained during the encryption/decryption of the S^{th} block of plaintext/ciphertext is used for calculating the seed for the LCG by $IC = X_{new}$.

3.3.1.8 STEP H

The ciphertext/plaintext (C_i / P_i) is thus constructed for one group of blocks at a time as explained in step F. The whole process of encryption/decryption continues till the plaintext/ciphertext is completely exhausted. The process of decryption is completely similar to the encryption process in this cipher – except for the Step F where (3.10) is used for encryption while (3.11) for decryption process.

3.3.2 Analysis of Pareek et al. Chaotic Block Cipher

Before undertaking the analysis of Pareek et al. chaotic block cipher, let us remind ourselves about one of the fundamental assumptions in cryptanalysis, first stated by A. Kirchoff in 1883 [Simmons 1992].

> "*The adversary knows all the details of the cryptosystem, including the algorithm and its implementation, except the secret key, on which the security of the cryptosystem must be entirely based*"

As for chaotic cryptosystems, the system parameters play a fundamental role as they are expected to act as the secret key. In what follows the following defects in Pareek et al. ciphers are explained:

1. Low sensitivity to plaintext
2. Reduced key space size
3. Vulnerable against the differential and known plaintext – attacks.

3.3.2.1 Low Sensitivity to Plaintext

In both conventional and chaotic cryptography [Stallings, 1998] it is well known that a properly designed and a good cryptosystem should be sufficiently sensitive with respect to both the changes in the plaintext and the secret key i.e. it should fulfill the avalanche property with respect to both the plaintext and secret key. For a given transformation to exhibit the avalanche property, an average of on one half of the output bits should change whenever a single input bit is complemented. However the Pareek et al. chaotic block cipher does not possess this property with respect to the plaintext. Looking into (3.10), it can be easily identified that one bit of change in P_i will result in only one bit of change in C_i. If the same key is used for the encryption, then $\lfloor X_{new} \times 10^5 \rfloor$ remains the same. The only case when the corresponding changes in P_i and C_i will not be equal is when the change in P_i is so large that it results in an extra modulo operation in (3.10).

3.3.2.2 Reduced Key Space Size

The security of a chaotic cryptosystem mainly relies on initial conditions (*ICs*) and control parameters. The initial condition in the proposed block cipher is derived from an external secret key. A 128-bit secret key that has 2^{128} possible combinations should relate to, at least, 2^{128} different initial conditions. In other words the algorithm to convert secret key to initial conditions must have uniform invariant measure[1]. Ideally different external secret keys must result in different *ICs* within the domain $(0,1)$ of the employed chaotic maps. In case of Pareek et al.'s cipher, it can be seen from equations (3.4) and (3.5) that the mapping process to an IC loses the positional significance of the characters of the secret key.

In order to illustrate this, let the secret key be formed by taking each character from the set $\kappa = \{0,\cdots,9,A,\cdots,F\}$. Consider a key, $K = 0123\cdots F$. This key results in some IC, say, x. If one swaps the positions of, say, 0 and 1 in the new key, it would also result in the same $IC = x$. In the considered example, the number of possible swaps are 16!. Hence, 16! different keys result for the same $IC = x$.

It can be seen that there are many other possible combinations of the characters that can be used to form the secret key K. In the example mentioned earlier, only those combinations are considered where all the 16 characters of the secret key are distinct. However, all other possible combination for the secret key K, that may consists of the same characters but having different positions, will also result in the same *IC*, say y, due to the fact that the positional significance of the characters in the key is not important at all. Hence, the key space size of the Pareek et al.'s ciphers is very much less than 2^{128}, - contrary to what is claimed by the authors.

3.3.2.3 Differential Attack – Proposed Attack 1

---

[1] An algorithm is said to have Uniform Invariant Measure if it has the property that if the input to the algorithm is uniform then the output of the algorithm should also be uniform.

All cryptanalytic attacks basically involve first searching for an identifiable nonrandomness property within the structure of the cryptographic algorithm [Phan & Siddiqi, 2006]. It allows one to differentiate between a black box containing the cryptographic algorithm and a black box containing the random permutation. This is known as the distinguishing property (or, simply, distinguisher) and further allows to successfully recovering the key (equivalent key) and the ciphered data.

The distinguisher varies from one cryptanalytic technique to another. However, the most common characteristics among the distinguishers for the various cryptanalytic techniques is the difference property between the pair of sequences (x, x'), where the difference Δx propagates with some probability p^*, through a cipher to result in a corresponding output pair of sequences (y, y'), with a specific difference Δy. In the proposed differential attack, based on the output difference Δy and knowing one corresponding plaintext sequence x, 2nd plaintext sequence x' can be recovered due to the insensitivity of the Pareek et al.'s cipher to the plaintext.

In what follows, a differential attack to break Pareek et al.'s chaotic block cipher is presented. In the proposed attack, the parameters or initial conditions of the chaotic cryptosystems, which serves as the secret key, are not recovered. Instead a differential is calculated from the two ciphertext sequences C_1 and C_2. It is shown that by using this differential and an a priori known plaintext sequence P_1, corresponding to C_1, 2nd plaintext sequence P_2 of the length M can be recovered.

Search for Distinguisher

Let P_1 and P_2 be the two different sequences consisting of the plaintext data, then, $\Delta P = P_{2i} - P_{1i}$ is the difference between the corresponding i^{th} block in the plaintext sequences, P_1 and P_2. The probability distribution of ΔP is p. In the next stage of the search, P_1 and P_2 are encrypted by Pareek et al. chaotic block cipher by using an arbitrary albeit same secret key, K, resulting in the two cipher sequences C_1 and C_2. $\Delta C_i = C_{2i} - C_{1i}$, constitutes the difference between the corresponding i^{th} blocks in the ciphertext sequences C_1 and C_2. The probability distribution of ΔC is c.

From the experimental analysis of ΔP and ΔC , it is found that 87-91% of the time, $\Delta P_i = \Delta C_i$, and only 09-13% of the time, $\Delta P_i \neq \Delta C_i$. When $\Delta P_i \neq \Delta C_i$, it is due to the modulo operator in (3.10), which is collectively applied on $(\alpha + \beta)$, where $\alpha = P_i$ and $\beta = \lfloor X_{new} \times 10^5 \rfloor$.

Fig. 3.1 displays the histogram of $\Delta P = P_{2i} - P_{1i}$, while Fig. 3.2 shows the histogram of $\Delta C = C_{2i} - C_{1i}$. As we know, the distribution of the plaintext data is usually non-uniform and encryption aims at making the distribution of the ciphered data as uniform as possible. Uniform distribution is the sign of the randomness. But looking at the histograms of $\Delta P = P_{2i} - P_{1i}$ and $\Delta C = C_{2i} - C_{1i}$ in Figs. 3.1 and 3.2 respectively, resemblance can be easily figured out between the two. Changes in plaintext are linearly related to changes in ciphertext. It goes to show that changes in plaintext do not lead to large changes in ciphertext. In other word, the said cipher of Pareek et al. is not sensitive to changes in plaintext. Only odd times, the corresponding differences ΔP_i and ΔC_i are not equal.

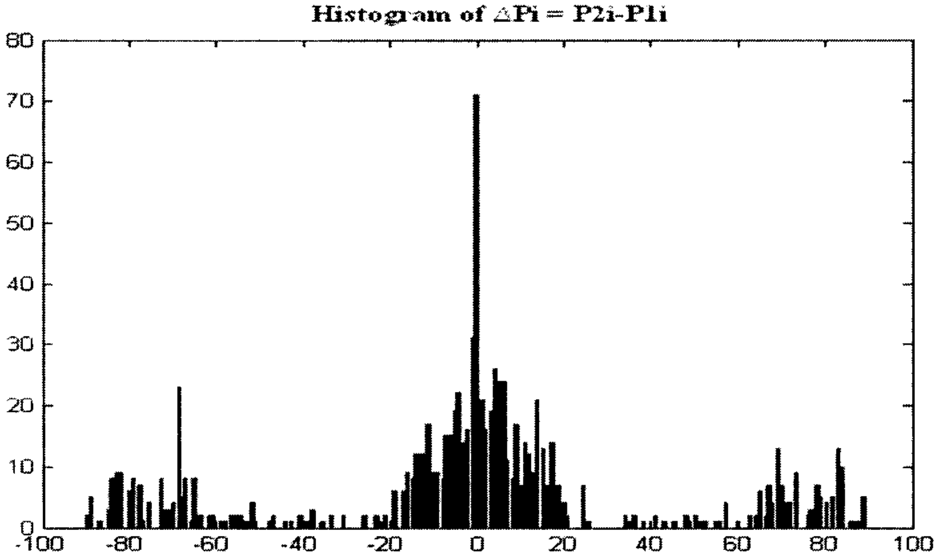


Fig. 3.1. The histogram for $\Delta P = P_{2i} - P_{1i}$

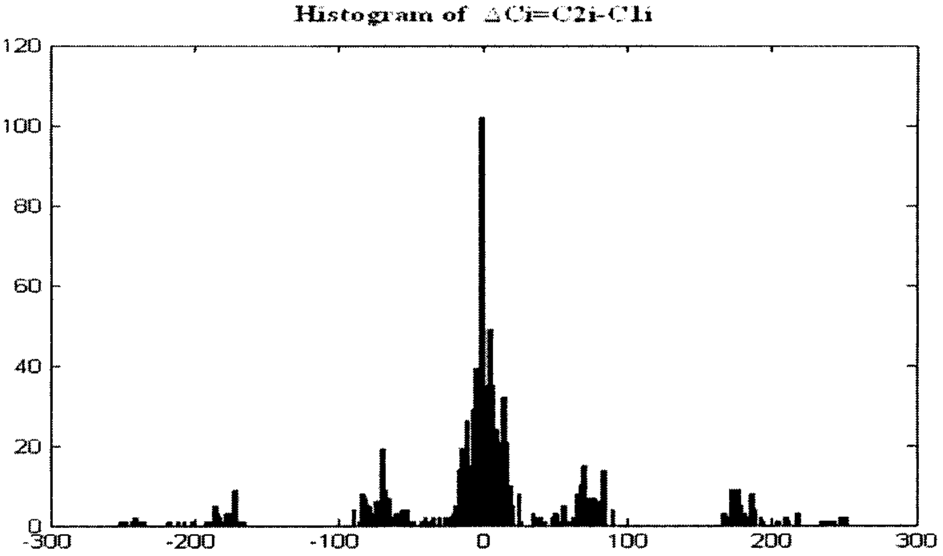


Fig. 3.2. The histogram for $\Delta C = C_{2i} - C_{1i}$:The resemblance between the Fig. 3.1 and Fig. 3.2 is evident. The number of times when $\Delta P_i = \Delta C_i$ is equal to 793 out of 905 blocks i.e. about 87.6243 %.

Recovery of Plain Data

Utilizing this distinguishing property in the chaotic block cipher of Pareek et al., it will be shown how the plain data can be recovered. Let the cryptanalyst know the plaintext P_1 and its corresponding cipher text C_1 of length M. If the cryptanalyst also has the access to another cipher sequence C_2, corresponding to the plaintext sequence P_2 of the length M , encrypted by an arbitrary albeit same secret key, then the proposed differential attack can be carried out in the following simple manner:

Step A. Compute the Differential

$$\Delta D_{21_i} = C_{2i} - C_{1_i} \quad 1 \le i \le M \tag{3.12}$$

where ΔD_{21_i} is the difference between the corresponding blocks in C_{2_i} and C_{1_i} consisting of M bytes.

Step B. Computing the Plaintext P_2

Given P_1, the plaintext P_2 can be computed sing the following equation:

$$P_{2i} = (P_{1_i} + \Delta D_{21_i}) \bmod 256 \quad 1 \le i \le M \tag{3.13}$$

where P_{2_i} is the i^{th} recovered plaintext block in the plaintext sequence P_2 of the length M.

Since ΔD_{21_i} is not always equal to ΔP_{21_i}, a 'modulo' operator is used as in (3.13) to make sure the correct recovery of the plaintext sequence P_2 as $(\Delta D_{21_i}) \bmod 256 = \Delta P_{21_i}$.

Tables 3.2 and 3.3, show the application of the proposed attack on the Pareek et al. chaotic block cipher. Table 3.2 shows the process of encrypting the plaintext sequence '*chaotic*' by the said cipher using the secret key '*3839384B4A44534457453233*'. The corresponding map used in the encryption is denoted by the map number N, while IC and IT represent the initial condition and number of iterations used for the corresponding chaotic map N, respectively. X_{new} is the output of the corresponding chaotic map, N, iterated by a number of times equal to IT. P_{1i} is the ASCII values of the corresponding i^{th} character in the plaintext sequence '*chaotic*', while C_{1i} is the corresponding ciphered value for P_{1i}. Table 3.3 shows the process of encrypting the plaintext sequence '*voltage*' by the said cipher using the secret key '*3839384B4A44534457453233*' and decrypting it by the proposed differential attack. From the Table 3.3, it can be seen that P_{2i} is equal to $(P_{1i} + \Delta D_{21i}) \bmod 256$.

Table 3.2. Encryption of a Specific Plaintext '*chaotic*' by Pareek et al.'s cipher: External Secret Key '*3839384B4A44534457453233*' (Expressed in Hexadecimal) is used

Plaintext Symbol	Map No. N	IC for map N	IT for map N	Output of map N X_{new}	P_{1i}	C_{1i}
C	3	0.1016	75	0.5039	99	57
H	3	0.5039	75	0.5780	104	47
A	3	0.5780	75	0.2078	97	141
O	1	0.1016	57	0.3387	111	189
T	2	0.1016	83	0.9699	116	82
I	2	0.9699	83	0.1381	105	91
C	2	0.1381	83	0.6459	99	177

Table 3.3. Encryption of a Specific Plaintext '*voltage*' by Pareek et al.'s cipher: External Secret Key '*3839384B4A44534457453233*' (Expressed in Hexadecimal) is used

Plaintext Symbol	P_{1i} (Taken from table 3.2)	P_{2i}	$D_{21i} = C_{2i} - C_{1i}$	$P_{2i}=(P_{1i}+ D_{21i}) mod\, 256$	C_{2i}
V	99	**118**	19	**118**	76
O	104	**111**	7	**111**	54
L	97	**108**	11	**108**	152
T	111	**116**	5	**116**	194
A	116	**97**	19	**97**	63
G	105	**103**	-2	**103**	89
E	99	**101**	2	**101**	179

3.3.2.4 Known-Plaintext Attack – Proposed Attack 2

In this scenario, P_1 and C_1 are known, and they are used for determining the chaotic stream corresponding to the secret key. The recovered chaotic key stream can then be used to recover any other plaintext completely, whose length is less than or equal to M - length of the chaotic key stream. The Pareek et al. cipher employs the "*modulo*" operator in (3.10), that is applied on the both the plaintext and extracted chaotic stream given by

$$(\alpha_i + \beta)\bmod 256, \qquad \text{where } \alpha = P_i \text{ and } \beta = \lfloor X_{new} \times 10^5 \rfloor$$

Here, it is shown how the chaotic stream can be extracted using the known plaintext attack. Let the adversary know the corresponding plaintext and ciphertext a priori. Starting with the following equation (3.14a),

$$C_i = P_i + \left(\lfloor X_{new} \times 10^5 \rfloor\right)\bmod 256 \tag{3.14a}$$

we will show that the plaintext P_i can be recovered. Let us rewrite (3.14a) as

$$X_i = C_i - P_i = \left(\left\lfloor X_{new} \times 10^5 \right\rfloor\right) \bmod 256 \tag{3.14b}$$

where X_i is the extracted chaotic key stream, which may or may not be the correct chaotic key stream. In order to ensure the correct value of X_i is recovered, the following routine is applied on X_i, where only two possibilities are checked.

$$\begin{aligned} &\textit{if } X_i < 0 \text{ then} \\ &X_i = X_i \bmod 256 \\ &\textit{else} \\ &X_i = X_i \end{aligned} \tag{3.15}$$

The output of the routine in (3.15) will result in the correct value of the chaotic key stream. Now, if the same external secret key is used for the encryption of more than one plaintext, then the extracted chaotic key stream can be used as an equivalent key for decrypting any ciphertext sequence having the length $L \leq M$. If the length $L \geq M$, then only M - bytes of the ciphered sequence can be recovered by using (3.16).

$$P_j = \left(C_j - X_i\right) \bmod 256, \tag{3.16}$$

where P_j is the recovered plaintext.

3.3.3 Fei et al. Chaotic Image Encryption Algorithm

Fei et al. recently proposed an encryption scheme especially for images in [Fei et al., 2005], based on multiple 3-D chaotic dynamical systems similar to Pareek et al.'s cipher. The proposed cipher employs the Lorenz, Chen, and LÜ chaotic systems given by (a) (b) and (c) in equation (3.17) respectively. In addition, initial conditions for the chaotic systems used in Fei et al.'s cipher are also acquired in a similar manner as Pareek et al.'s cipher. This leads to reduced key space size – a symptom from which Pareek et al.'s cipher also suffered.

$$\begin{array}{ccc} x' = a(y-x) & x' = a(y-x) & x' = a(y-x) \\ y' = cx - xz - y & y' = (c-a)x - xz + cy & y' = -xz - cy \\ z' = xy - bz & z' = xy - bz & z' = xy - bz \\ \text{(a)} & \text{(b)} & \text{(c)} \end{array} \qquad (3.17)$$

3.3.3.1 Brief Description of Fei et al. Image Cipher

The chaotic signals are mainly dependent on the initial conditions (x_0, y_0, z_0) and the iteration time k. In Fei et al.'s cipher, the initial conditions (x_0, y_0, z_0), iteration time, k, and condition number, f, are generated by mapping the external keys. According to the value of the condition number, f, a certain chaotic system is selected from the three systems, iterated k times, and used for acquiring (x_k, y_k, z_k). Chaotic key stream is extracted from (x_k, y_k, z_k) according to some operations, which is then used to encrypt the R, G, B values of a pixel. The next iteration time of the mixed chaotic dynamical systems depends on the previous ciphered image pixel and the previous iteration time. In what follows, the sub-algorithm used for acquiring initial condition (x_0, y_0, z_0), first iteration time k_1 and condition number, f, is described in some detail.

3.3.3.2 Sub-algorithm Used for Extracting of ICs from External Secret Key

In Fei et al. cipher, sub-algorithm used for extracting *ICs* is explained in Step 1 of [Fei et al., 2005]. Since the systems in the proposed cipher are three dimensional, hence three real numbers $U, V, W \in [0,1]$ are acquired by the following formulas to act as the *ICs* for the multiple chaotic dynamical systems.

Let T_1, T_2 and T_3 be derived from external key $K = [K_1\ K_2 \cdots K_{16}]$ as:

$$T_1 = K_1 + K_2 + \cdots\cdots + K_{16} \tag{3.18}$$

$$T_2 = \left((K_1)_2 \oplus (K_2)_2 \oplus \cdots\cdots \oplus (K_{16})_2)\right)_{10} \tag{3.19}$$

$$T_3 = T_1 * T_2 \tag{3.20}$$

where $(K_i)_2$ is binary representation of K_i. The parameters U, V and W are now defined as:

$$U = {(T_1 \bmod 256)}\big/{256} \tag{3.21}$$

$$V = {(T_2 \bmod 256)}\big/{256} \tag{3.22}$$

$$W = {(T_3 \bmod 256)}\big/{256} \tag{3.23}$$

Now the first iteration time, k_1, is defined as:

$$k_1 = 5 + \left(\lfloor (T_1 + T_2 + T_3)/3 \rfloor \bmod 25\right) \tag{3.24}$$

Lastly the control number, f, is defined as:

$$f = T_3 \bmod 3 \tag{3.25}$$

3.3.3.3 Reduced Key Space Size

The initial conditions (x_0, y_0, z_0), first number of iterations, k_1, and control number, f, are acquired by mapping the external secret key according to the algorithm mentioned in section 3.3.3.2. However, the mapping process loses the positional significance of the each character in the secret key similar to the Pareek et al.'s cipher. It can be confirmed from the equations (3.18), (3.19) & (3.20) that during acquiring the values for the intermediate variables $T_1, T_2 \text{ and } T_3$ from the external secret key, the positional significance of the characters in the key is rendered unimportant. Due to this fact, different keys result in the same values for the intermediate variables $T_1, T_2 \text{ and } T_3$, resulting in the same values for the variables $U, V \text{ and } W$ that serve as the initial condition for the chaotic systems. Hence the actual key space size of the Fei et al.'s cipher is less than 2^{128}, contrary to what is claimed by the authors. The detailed explanation of the said symptom can be found in the section 3.3.2.

In order to illustrate this, the said symptom is also practically demonstrated in Fig.3.3. An image is encrypted by this cipher using the secret key "<*&*%&*%\;@#%87?", shown in frame (A) of the Fig.3.3. Frame (B) of the Fig.3.3 shows the successfully decrypted image with the wrong key "?78%#@;\%*&%*&*<", which is totally opposite of the encrypting key.

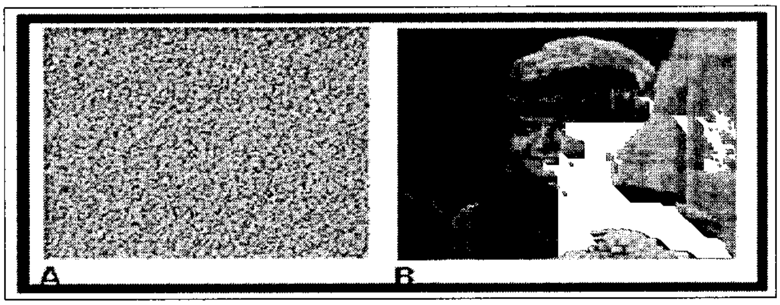


Fig. 3.3. Practical demonstration of loss of positional information in Fei et al.' image cipher: Image in frame (A) encrypted with secret key ''<*&*%&*%\;@#%87?', Image in frame (B) is decrypted with the key '?78%#@;\%*&%*&*<'

3.3.3.4 Brute Force Attack

A brute force attack is the method of breaking a cipher by trying every possible key. The quicker the brute force attack, the weaker the cipher. Feasibility of brute force attacks depends on the key space size of the cipher and on the amount of computational power available to the attacker.

In Fei et al.'s cipher, it is identified that even if the positional significance of the each 8-bit character in the secret key is retained during the extraction of ICs, the said cipher has very small key space size. The key space size of the proposed cipher is only 256^3, that can be easily explained. By observing (3.21), (3.22) and (3.23), one can see that there are only 256 possible values for each U, V and W. Hence the key space is only 256^3. After guessing the values of U, V and W, the values for k_1 and f can be guessed very easily, as the value of k_1 is in between 5 and 29 and the value of f is either 0, 1 or 2.

3.3.4 Pareek et al.'s Image Encryption Scheme

In this section, we first present Pareek et al.'s recent most image encryption scheme based on chaotic logistic map [Pareek et al., 2006] and then perform its detailed analysis. Pareek et al.'s image encryption scheme consists of the following main steps:

3.3.4.1 Step 1

The proposed image cipher utilizes an external secret key, 80-bit long. Further, the secret key is divided into blocks of 8-bit each, referred to as session keys.

$$K = k_1 k_2 \ldots k_{20} \text{(in hexamdecimal)}, \tag{3.26}$$

$K = K_1 K_2 \ldots K_{10}$ (in ASCII),

where, each K_i represents one 8-bit block of the secret key i.e. session key.

3.3.4.2 Step 2

In the proposed algorithm, two chaotic logistic maps are employed to achieve the goal of image encryption given as follow:

$$X_{n+1} = 3.9999 X_n (1 - X_n), \tag{3.27}$$

$$Y_{n+1} = 3.9999 Y_n (1 - Y_n), \tag{3.28}$$

The initial conditions (X_0 and Y_0) for these maps are calculated using some mathematical manipulations on session keys. The logistic map in (3.27) helps in deriving the initial condition for the 2nd logistic map in (3.28). The initial condition for the first logistic map is extracted from the external secret key using some mathematical manipulations on the session keys, whose detail can be found in Step 3, 4, and 5 of the Pareek et al.'s cipher [Pareek et al., 2006].

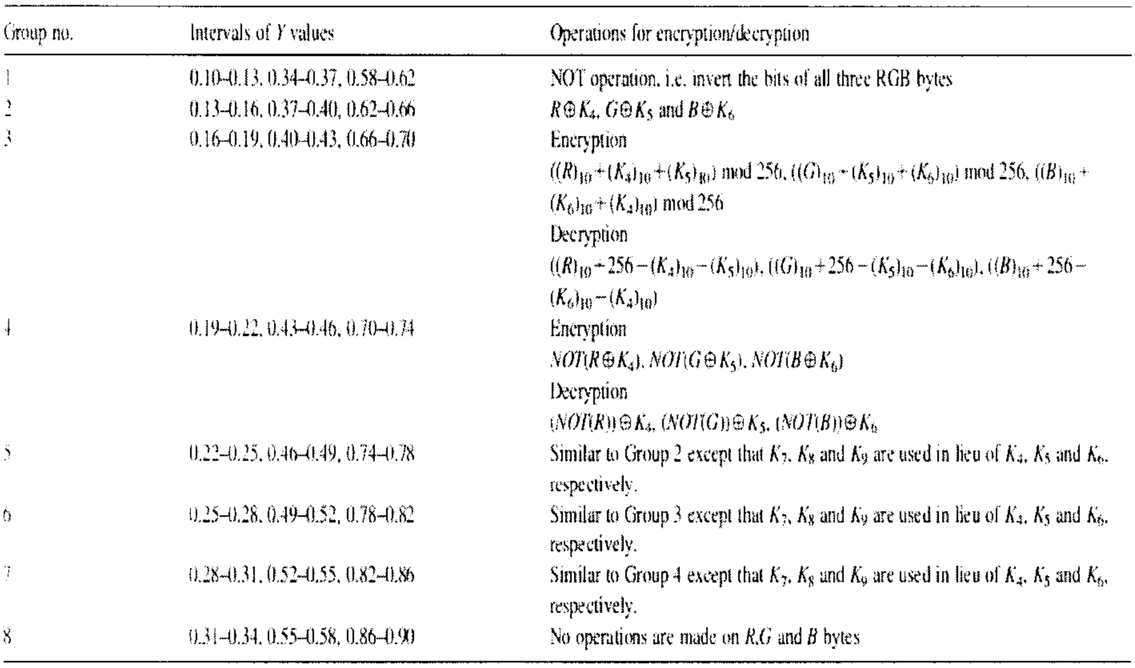
3.3.4.3 Step 3

Read three consecutive bytes from the image file. These three bytes represent the value of the red, green and blue (RGB) color respectively and together form a single pixel of the image.

3.3.4.4 Step 4

The range [0.1, 0.9] is divided into 24 non-overlapping intervals and arranged into eight different groups. Different types of operations are assigned corresponding to each of these groups. In Fig. 3.4, the interval ranges for the groups and their corresponding operations are given. Further, a second logistic map is iterated once using the initial condition Y_0 obtained in Step 5 of [Pareek et al., 2006]. The outcome of the second logistic map decides which operation is to be performed for encryption/decryption of red, green and blue (RGB) color bytes.

This step is repeated $(K_{10})_{10}$ (i.e. the decimal equivalent of 10th session key) times. Finally, the encrypted bytes of red, green and blue colors are written in a file i.e. encryption of a pixel is achieved. The steps 3 and 4 are repeated for next 15 pixels of the image file.



Group no.	Intervals of Y values	Operations for encryption/decryption
1	0.10–0.13, 0.34–0.37, 0.58–0.62	NOT operation, i.e. invert the bits of all three RGB bytes
2	0.13–0.16, 0.37–0.40, 0.62–0.66	$R \oplus K_4$, $G \oplus K_5$ and $B \oplus K_6$
3	0.16–0.19, 0.40–0.43, 0.66–0.70	Encryption $((R)_{10}+(K_4)_{10}+(K_5)_{10})$ mod 256, $((G)_{10}+(K_5)_{10}+(K_6)_{10})$ mod 256, $((B)_{10}+(K_6)_{10}+(K_4)_{10})$ mod 256 Decryption $((R)_{10}+256-(K_4)_{10}-(K_5)_{10})$, $((G)_{10}+256-(K_5)_{10}-(K_6)_{10})$, $((B)_{10}+256-(K_6)_{10}-(K_4)_{10})$
4	0.19–0.22, 0.43–0.46, 0.70–0.74	Encryption $NOT(R \oplus K_4)$, $NOT(G \oplus K_5)$, $NOT(B \oplus K_6)$ Decryption $(NOT(R)) \oplus K_4$, $(NOT(G)) \oplus K_5$, $(NOT(B)) \oplus K_6$
5	0.22–0.25, 0.46–0.49, 0.74–0.78	Similar to Group 2 except that K_7, K_8 and K_9 are used in lieu of K_4, K_5 and K_6, respectively.
6	0.25–0.28, 0.49–0.52, 0.78–0.82	Similar to Group 3 except that K_7, K_8 and K_9 are used in lieu of K_4, K_5 and K_6, respectively.
7	0.28–0.31, 0.52–0.55, 0.82–0.86	Similar to Group 4 except that K_7, K_8 and K_9 are used in lieu of K_4, K_5 and K_6, respectively.
8	0.31–0.34, 0.55–0.58, 0.86–0.90	No operations are made on R,G and B bytes

Fig.3.4. Intervals and their corresponding operations used in Pareek et al.'s image cipher: Different groups of non-overlapping intervals of Y values and corresponding operations for image encryption and decryption

Here, K_i's are the session keys, R, G and B, are the red, green and blue levels of image pixels, respectively.

3.3.4.5 Step 5

After encryption of a block of 16 pixels of image file, the session keys K_1 to K_9 are modified as follows:

$$(K_i)_{10} = ((K_i)_{10} + (K_{10})_{10}) \bmod 256, \quad (1 \le i \le 9). \tag{3.29}$$

3.3.5 Analysis of Pareek et al.'s Image Encryption Scheme

In this section, Pareek et al.'s recently proposed image encryption scheme [Pareek et al., 2006] is analyzed. It is shown that Pareek et al.'s encryption scheme does not fulfill real time requirements for image encryption, contrary to the claim made by Pareek et al. and is insecure against the proposed differential attack. It also does not fulfill the avalanche property at the level of image.

3.3.5.1 On the Question of its Suitability for Real Time Encryption of Images

Pareek et al. claim that the proposed scheme in [Pareek et al., 2006] is suitable for real time secure encryption of the digital images. They reported average encryption times for different sizes of color images in the Table 4 of [Pareek et al.'s, 2006]. The results are very encouraging and show that the proposed scheme is suitable for real time secure image transmission. However, from our analysis of the Pareek et al.'s scheme in [Pareek et al., 2006], it is identified that the proposed scheme is not suitable for the real time encryption. It is because the number of iterations of the 2nd logistic map for the encryption of just one pixel is dependent on $(K_{10})_{10}$, whose values range between 0 & 255. In fact, the value of $(K_{10})_{10}$ can not be 0, as then the number of iterations will be zero, which means no encryption at all. However let us assume the value of $(K_{10})_{10}$ is anything between 1 & 255, it shows that the minimum and maximum number of iterations for encryption of just one pixel is 1 and 255, respectively. However, the value of $(K_{10})_{10}$ can not be very low, because that would mean the security level is very low. Hence, encryption of an image where number of iterations used for any pixel is sometimes very low and another time very high, is not a very secure design. In addition, assuming an average $(K_{10})_{10}$ for all pixels to be, say, 20, it amounts to 20 rounds of encryption, which is also not suitable for real time image encryption where images themselves are very bulky.

However in [Pareek et al.'s, 2006], they carried out all of their analysis with the following two keys: 1) '*ABCDEF0123456789FF05*'; 2) '*ABCDEF0123456789FF06*', where the value of $(K_{10})_{10}$ is 5 and 6 respectively. Hence in our opinion, the reported time in Table 4 of [Pareek et al.'s, 2006], may only be based on the value of $(K_{10})_{10}$ for the above mentioned two secret keys. Hence claiming it for all values of $(K_{10})_{10}$ will not be correct and the average encryption speed for different values of $(K_{10})_{10}$ will be very low, compared to the average encryption speed reported in the Table 4 of [Pareek et al., 2006].

3.3.5.2 Differential Attack

In this section, it will be shown that it is very easy to partially recover the plain-image, with the help of a mask image, f_m, which can be used as an equivalent key for the secret key K. It will be shown that this mask image can partially recover low quality image, whose size is not larger than the size of, f_m.

3.3.5.2.1 Obtaining f_m from Two Ciphered Images

Assume that an $M \times N$ plain-image f_K and its corresponding cipher-image $f_K^{'}$ are known to an attacker. Then if the attacker wants to decrypt another cipher image $f^{'}$, he can get f_m by simply XORing the two ciphered images $f_K^{'}$ and $f^{'}$, pixel by pixel:

$$f_m(i,j) = f_K^{'}(i,j) \oplus f^{'}(i,j), \text{ where } 0 \le i \le M-1 \text{ and } 0 \le j \le N-1. \tag{3.30}$$

With the mask image f_m, the attacker tries to recover the plain-image by XORing the mask image and known plain-image f_K pixel by pixel:

$$f_R(i,j) = f_K(i,j) \oplus f_m(i,j), \text{ where } f_R \text{ is the recovered image.} \tag{3.31}$$

Let $f_{m-\text{plain}}$ be the mask image corresponding to the two plain-images and is equal to:

$$f_{m\text{-Plain}}(i,j) = f_K(i,j) \oplus f(i,j). \tag{3.32}$$

From $(3.30) = (3.32)$, it can be seen that then (3.31) holds and will result in the correct recovery of the corresponding pixel value; otherwise (3.31) is generally not true i.e. $f_R(i,j)$ will not be equal to $f(i,j)$.

To show the real performance of the mask image f_m, several images are encrypted by Pareek et al.'s cipher using the secret key '*ABCDEF0123456789FF05*', which they used throughout the analysis of their cipher in [Pareek et al., 2006]. Figs. 3.5., 3.6, 3.7, and 3.8, shows the performance of the mask image f_m. Although, it is not possible to fully recover the plain image with f_m, however the major contents of the images are still visible, as shown in Fig. 3.8. In order to show, why f_m is able to recover some of the contents of a plain-image, in what follows, we show the influence of various operations used in the encryption of a pixel on difference $\Delta x = x \oplus x^*$, where x and x^* represents two pixel values in two different images and Δx their XOR difference.

Proposition 1: Key XORing has no affect on $\Delta x = x \oplus x^*$.

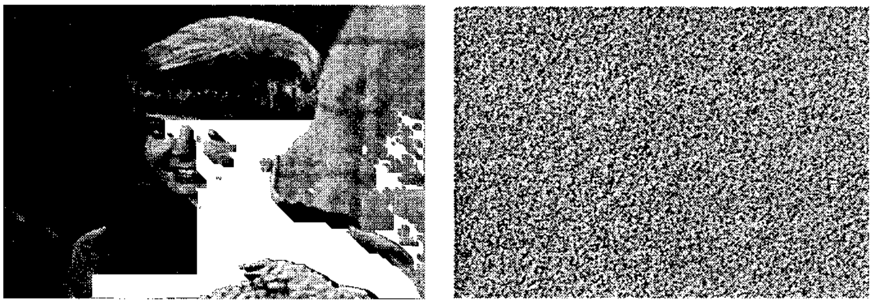
Proof: Let the same secret key K be used for XORing x and x^*, then $\Delta x = (x \oplus K) \oplus (x^* \oplus K) = x \oplus x^*$. Hence proved.

Proposition 2: "*Not*" operation does not affect $\Delta x = x \oplus x^*$.

Proof: Let $x = a_7 a_6 a_5 \cdots a_0$ and $x^* = b_7 b_6 b_5 \cdots b_0$, represent the binary form of x and x^*. After the "*Not*" operation, let x and x^* be denoted by $\widetilde{x} = \widetilde{a}_7 \cdots \widetilde{a}_0$ and $\widetilde{x}^* = \widetilde{b}_7 \cdots \widetilde{b}_0$. From the property of the XOR operator, $a_i \oplus b_i = \widetilde{a}_i \oplus \widetilde{b}_i = c_i$, hence $\Delta x = (a_7 \oplus b_7) \cdots (a_0 \oplus b_0) = (\widetilde{a}_7 \oplus \widetilde{b}_7) \cdots (\widetilde{a}_0 \oplus \widetilde{b}_0)$.

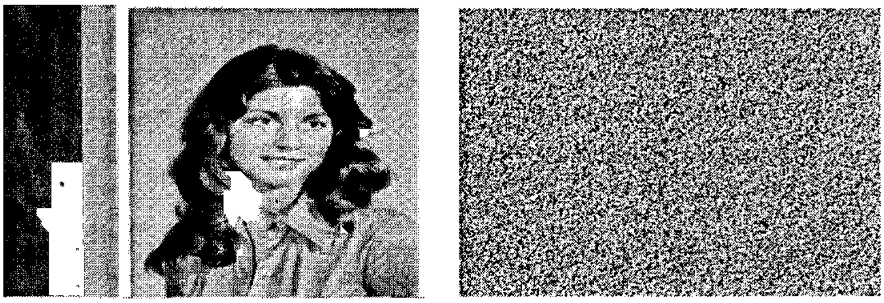
Proposition 3: "*modulo*" operator may or may not affect $\Delta x = x \oplus x^*$.

Proof: We carried out this analysis for determining the influence of the "*modulo*" operation on different values of input and key values, for the number of iterations equal to 255. The pseudo code for the analysis so carried out is given in Fig. 3.9. It is found that for almost 25% of the key values "*k*", "*modulo*" operator is unable to change the value of Δx. It is also observed that for those values where $\Delta x \neq \Delta y(i,k)$, the corresponding absolute difference is $|\Delta x - \Delta y(i,k)| \cong \Delta x$. In particular for $\Delta x = 3$, it is noticed that if $\Delta x \neq \Delta y(i,k)$, then $\Delta y(i,k)$ is often equal to 1. It means that there is only one bit of error or hamming distance between the two is 1. The same phenomenon was observed for all other values of Δx. Hence, for an image, where a strong correlation exists between the adjacent pixels, even if one bit of error occurs, still main contents of image will be visible.



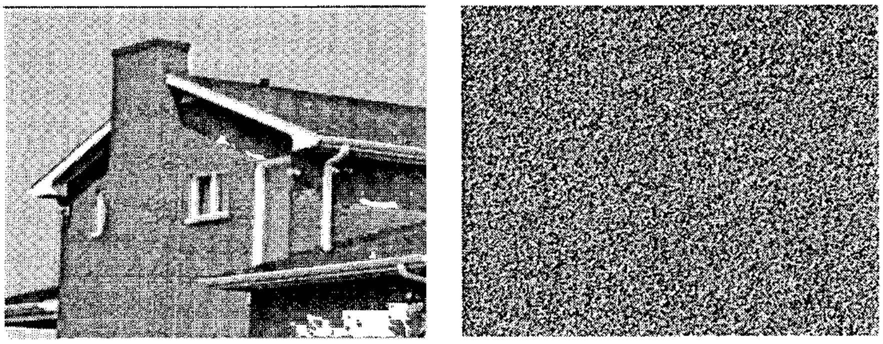
a) The Known plain image f_{Barbra} **b)** The cipher image f'_{Barbra}

Fig. 3.5. f_{Barbra} and f'_{Barbra}: One Known-Plain Image of the size (256 × 256) and its cipher image.



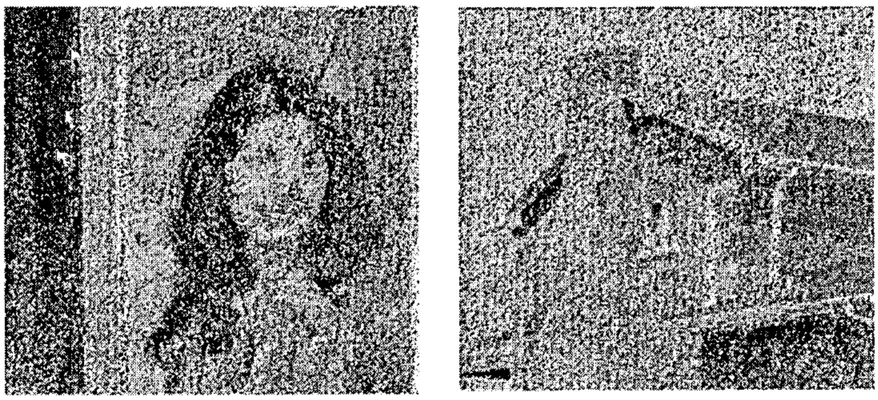
a) The plain image f_{Girl} **b)** The cipher image f'_{Girl}

Fig. 3.6. f_{Girl} and f'_{Girl}: A Plain-Image unknown to the attacker, 'Girl', of the size (256 × 256) and its cipher image.



a) The plain image f_{House} **b)** The cipher image f'_{House}

Fig. 3.7. f_{House} and f'_{House} : A Plain-Image unknown to the attacker, 'House', of the size (256 × 256) and its cipher image.



a) The recovered image $f_{\text{R-Girl}}$ **b)** The recovered image $f_{\text{R-House}}$

Fig.3.8. Recovering Plain Images: Recovering plain-images with f_m (derived from f' and f'_{Barbra}) and one known plain-image f_{Barbra}

The histograms of a mask image (consisting of differentials) obtained from plain-images, "Barbara" and "Girl" is shown in Fig. 3.10, while the histogram of mask image (consisting of differentials) obtained from the ciphered-images of "Barbra" and "Girl" is shown in Fig. 3.11. It is surprisingly seen that the encryption performance is much better than expected: about 60.99% of the corresponding differentials in $f_m(i,j)$ and $f_{m\text{-plain}}(i,j)$ are equal; hence about 60.99% of pixels are correctly recovered.

In order to further improve the visual quality of the recovered plain-images, some noise reducing techniques can be used to further reduce the recovery of errors. Enhanced version of $f_{\text{R-Girl}}$ and $f_{\text{R-House}}$, are shown in Fig. 3.12 (a) (b), where noise is removed by 5×5 median filter. It can be seen that visual quality of $f_{\text{R-Girl}}$ and $f_{\text{R-House}}$, is enhanced.

```
Function modulo (difference)
$\Delta x$ =difference;
for $i$=0 step 1 to 255
            $x = i$.
            $x^{*} = x + \Delta x$.
    for $k$=0 step 1 to 255
            for j=0 step 1 to 255
                    $x = (x + k) \bmod 256$.
                    $x^{*} = (x^{*} + k) \bmod 256$.
            end for
            $Y_1(i,k) = x$.
            $Y_2(i,k) = x^{*}$.
            $\Delta y(i,k) = x \oplus x^{*}$.
    end for
end for
```

Fig. 3.9. Pseudo code for investigating the affect of "*modulo*" operator on Δx : Pseudo code for checking the influence of each 8-bit key value 'k' on the input values x and x^*, with the corresponding difference Δx, and number of iterations "j" equal to 255.

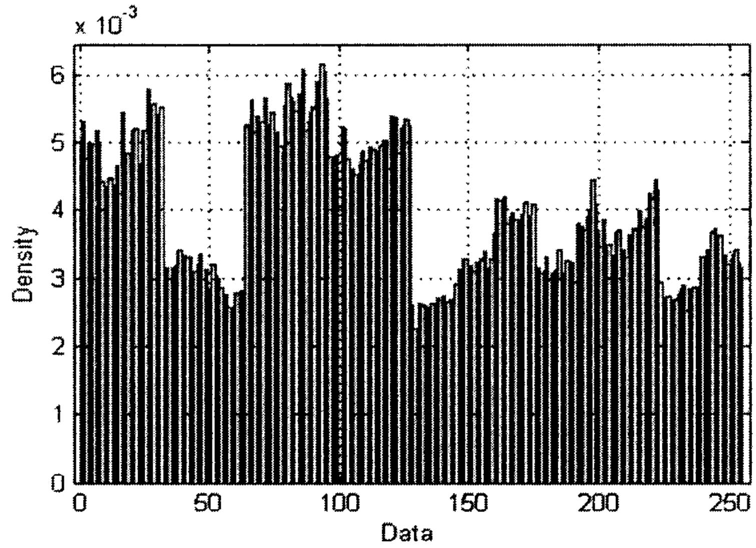


Fig. 3.10. Histogram of mask image obtained from plain images i.e. $f_{m\text{-plain}}(i,j) = f_{\text{Barbra}}(i,j) \oplus f_{\text{Girl}}(i,j)$.

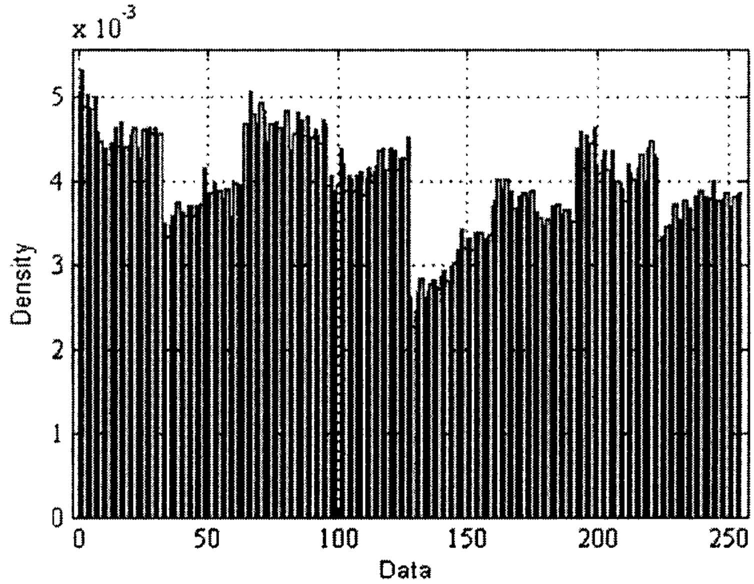


Fig. 3.11. Histogram of mask image obtained from ciphered images: i.e. $f_m(i,j) = f'_{\text{Barbra}}(i,j) \oplus f'_{\text{Girl}}(i,j)$.

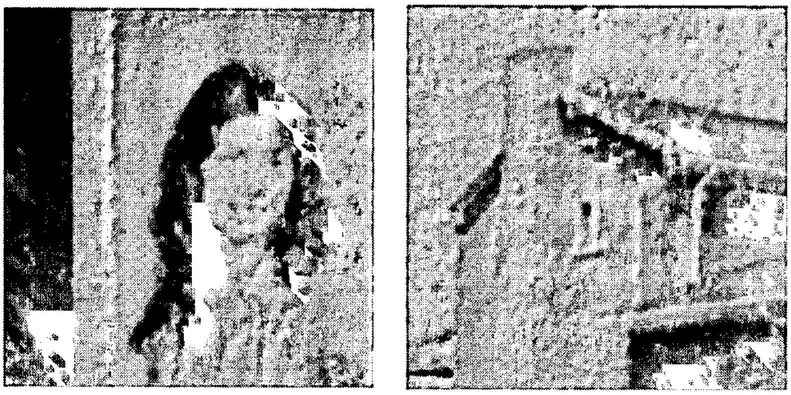


Fig. 3.12. Enhancing the recovered plain-images $f_{\text{R-Girl}}$ and $f_{\text{R-House}}$, with a 5×5 median filter

3.3.5.3 Defect in Pareek et al.'s Encryption Scheme

Before enumerating the defect in Pareek et al.'s image encryption scheme, we first present the proof of the following propositions.

Proposition 4: XOR Operation $\oplus$ is its own inverse if the same secret key K is used twice.

Proof: The proof of the above proposition is straight forward.

Proposition 5: "Not" is its own inverse.

Proof: Let X be an 8-bit value, which can be represented in its equivalent binary form as: $X = a_7 a_6 a_5 \cdots a_0$, then after the application of "Not" operation, $Y = Not(X) = \tilde{a}_7 \tilde{a}_6 \tilde{a}_5 \cdots \tilde{a}_0$.

Now the application of "Not" operation on Y will result in X i.e. $Not(Y) = \tilde{\tilde{a}}_7\tilde{\tilde{a}}_6\tilde{\tilde{a}}_5 \cdots \tilde{\tilde{a}}_0 = X$, as $\tilde{a}_i = b_i = \tilde{b}_i = a_i$. Hence proved.

Proposition 6: Combination of "Not" and XOR operations, are also their own inverse under the same secret key K.

Proof: Let X and Y be the input and output of a function $f(X) = Not(X \oplus K_i) = Y$, where K_i is 8-bit block from the secret key K. Alternatively, $f(X)$ can also be written as: $f(X) = Not(X) \oplus K_i = Y$. Now $f(Y) = Not(Y \oplus K_i) =$ $Not(Not(X \oplus K_i) \oplus K_i) =$ $Not(Not(X)) \oplus K_i \oplus K_i$ $= Not(Not(X))$. since "Not" is its own inverse from proposition 5, $f(Y) = Not(Not(X)) = X$. Hence proved.

The above propositions are proved in order to bring attention to a symptom in the structure of the Pareek et al.'s cipher, where operations are being selected dynamically on the basis of the output of the 2nd logistic map. In Pareek et al.'s scheme, secret key K is internally changed after encrypting 16 pixels. The number of iteration used for encryption of a pixel are equal to $(K10)_{10}$. The values of secret key blocks K_i remain constant during all iterations $(K10)_{10}$, used for encrypting a pixel. Now it is equally possible that the output of the logistic map will fall consecutively two or more times in the same group of intervals defined in Fig. 3.4. Suppose the output of the logistic map falls consecutively two times in group 1 (corresponding operation is "Not"), then the pixel value will go out without encryption for that two iterations of logistic map as "Not" operation is its own inverse. Same argument is true (from propositions 4, 5 and 6) for the operations related to groups 2, 4, 5 and 7, where either XOR, Not or combination of XOR and Not operations are used for encrypting a pixel value.

Hence for all operations except '*modulo*' in Pareek et al.'s image cipher, if they are executed even number of times consecutively, will have no affect on pixel value as they are their own inverse. This means that for those rounds, pixels will go out without encryption.

3.3.5.4 On the Sensitivity to Change in Pixel

In section 3.2 of [Pareek et al., 2006], Pareek et al.'s claims that the proposed scheme is highly sensitive to change in pixels and effect of change in one pixel is diffused over rest of pixels in the image. The Number of Pixel Change Rate (NPCR) is calculated for two encrypted images C1 and C2, whose corresponding original images have only one-pixel difference. Pareek et al.'s reports NPCR of about 99%. However, from the structure of the proposed scheme, it is clearly visible that encryption of a pixel is not dependent on the rest of pixels in the image. Hence their claim about NPCR of 99% is not true.

3.4 Discussion

From the cryptanalysis of Pareek et al.'s and Fei et al.'s ciphers in [Pareek et al., 2005] [Pareek et al., 2006] [Fei et al., 2005] in the proceeding sections, the following observations can be made.

3.4.1 Positional Significance of the Each Character in External Secret Key

The positional significance of each character in the secret key during the mapping of external secret key must be retained. However in Pareek et al. and Fei el al. ciphers [Pareek et al., 2005] [Fei et al., 2005], positional significance of the each character in the secret key is not lost during the mapping of the secret keys to ICs. Due to this lack of the positional information of the each character during the mapping process, different combinations of the secret keys having the same character results in the same *IC*. Hence the key size of the proposed ciphers is less than 2^{128} that is claimed by Pareek et al. and Fei et al. in [Pareek et al., 2005] [Fei et al., 2005].

3.4.2 Effect of Mapping on the Key Space Size

The key space size of a cipher should be analyzed in detail after the mapping process. Since in Pareek et al. and Fei et al. ciphers, an external secret key is employed that is mapped to an *IC*. The key space size of the secret key should correspond to the same size in *ICs*. In both ciphers, the length of the secret key is 128-bit which is further divided into 8-bit blocks. Based on the length of the secret key, it is claimed by Pareek et al. and Fei et al. in [Pareek et al., 2005] [Fei et al., 2005] respectively that the key space size is 2^{128}. However, in both Pareek & Fei cipher, key space is not analyzed after the mapping process and it is considerably small.

3.4.3 Sensitivity to the Plaintext

An essential condition for a cipher is the conformance to the avalanche property with respect to the plaintext data i.e. a single bit change in input plaintext must result in more than one bit change and preferably in all bits. Some of the researchers have referred to the above property as error propagation property or Strict Avalanche Criterion (SAC). For a cryptographic function to satisfy the SAC, each output bit should change with a probability of half whenever a single input bit is complemented. SAC is considered to be an important property in the design of the S-boxes. If a cipher lacks the said property, then it can be easily cryptanalyzed using chosen plaintext (differential) and known plaintext-attacks as demonstrated with the Pareek et al. ciphers in [Pareek et al., 2005] [Pareek et al., 2006].

3.4.4 Chaotic Key Stream Extraction

It is observed that in the cryptanalysis of most of the chaotic ciphers, chaotic key stream is extracted using the known and chosen plaintext attacks. The recovered key stream can then be reused to decrypt any ciphertext having the length less than or equal to the length of the chaotic key stream. Hence the cryptosystems should be so designed that even if the adversary is able to extract the chaotic keys stream, he is not able to recover the remaining plaintext. In Fei et al. cipher, it is observed that different chaotic key stream is used for the different plain images even if the same external secret key is employed for the encryption of the multiple images. It is done by ciphertext feedback. The inclusion of the ciphertext feedback makes it secure otherwise it would also suffer from the same weaknesses as Pareek et al. ciphers. Hence the inclusion of the internal (based on the previous plaintext) or external (based on the previous ciphertext) feedbacks can be an effective mechanism to designing ciphers.

3.5 Summary

In this chapter, three recent chaotic encryption schemes are cryptanalyzed. A number of security problems pertaining to these ciphers have been identified and reported in this chapter. It is shown that these ciphers are not suitable for the secure real time encryption of the digital images.

4

CHAPTER

Proposed Encryption Schemes

4.1 Introduction

In this chapter, two hybrid encryption schemes – Hybrid Chaotic Image Encryption Scheme (HyChIES) and Hybrid Chaotic Encryption Scheme (H-CES), are described. HyChIES qualifies as a complete image encryption scheme, whereas H-CES works as a 128 bit block cipher similar to AES.

In the last decade many encryption schemes based on chaos have been proposed and have attracted due attention for encrypting the information. Chaos and cryptography have some common features- the most prominent being sensitivity to small changes in variables and parameters. However, an important difference between chaos and cryptography lies in the fact that systems used in chaos are defined on real numbers, while cryptography deals with systems defined on finite sets of integers.

A look at most chaos based cryptosystems suggests that the apparently random-like chaotic sequence masking the plain image can not hide the image very well. The image can be easily recovered with relatively low computational load by the masters of cryptography. Chaos-based cryptographic schemes using chaotic masking is still in its infancy with many algorithms and cryptanalysis methodologies still under development. From our study, it is seen that the security of the chaotic ciphers proposed by various researchers is suspect against the standard techniques of cryptanalysis. For example, the encryption algorithm proposed in [Habutsu et al., 1991] is proved insecure in [Biham, 1991], showing that the algorithm can be broken using known-plaintext attack. In earlier chapter, we also showed two of the recent chaotic algorithms to be vulnerable against the known plaintext attacks. We have found that, on careful analysis, most of the chaotic ciphers prove weak against the current prevailing cryptanalytic techniques and do not possess the S-box equivalent security that the traditional ciphers can provide.

In what follows, HyChIES and H-CES are described in detail. However, before describing HyChIES and H-CES, we introduce some background on AES and S-box.

4.2 Background

4.2.1 Description of AES

The Advanced Encryption Standard (AES) specifies Federal Information Processing Standards (FIPS) - cryptographic algorithm approved by FIPS, that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can efficiently encrypt (encipher) and decrypt (decipher) any bit stream carrying information. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. The block diagram of AES for one round is shown in Fig.4.1.

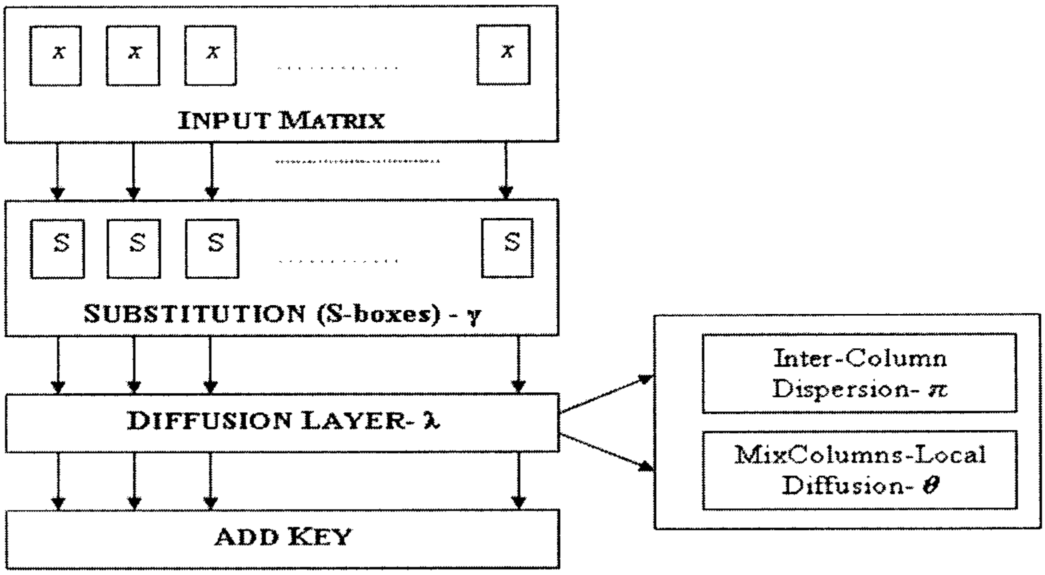

x
INPUT MATRIX
S
SUBSTITUTION (S-boxes) - γ
DIFFUSION LAYER- λ
ADD KEY
Inter-Column Dispersion- π
MixColumns-Local Diffusion- θ

Fig. 4.1. Block Diagram of AES for one round

4.2.1.1 AES Cipher

AES cipher consists of the following transformations, which are applied repeatedly:

- SubBytes ( ) Transformation
- ShiftRows ( ) Transformation
- MixColumns ( ) Transformation
- AddRoundKey ( ) Transformation

Pseudo Code for the AES Cipher is given in Fig. 4.2. The detailed description of AES can be found in [http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf].

```
Cipher (byte in [4*Nb], byte out [4*Nb], word w [Nb* (Nr+1)])

begin
    byte state [4, Nb]

    state = in

    AddRoundKey ( state, w [0, Nb-1])

    for round = 1 step 1 to Nr-1
        SubBytes (state)
        ShiftRows (state)
        MixColumns (state)
        AddRoundKey (state, w [ round * Nb, (round+1)* Nb-1])
    end for

    SubBytes (state)
    ShiftRows (state)
    AddRoundKey (state, w [Nr* Nb, (Nr+1) * Nb-1])


    out = state
end
```

Fig. 4.2. Pseudo Code for the AES Cipher
[http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf]

4.2.1.1 AES Inverse Cipher

For decryption, the inverse cipher is used, which consists of the following individual transformations:

- InvShiftRows ( ) Transformation
- InvSubBytes ( ) Transformation
- InvMixColumns ( ) Transformation
- AddRoundKey ( ) Transformation

Pseudo Code for Inverse Cipher is given in Fig. 4.3.

```
InvCipher (byte in [4*Nb], byte out [4*Nb], word w [Nb* (Nr+1)])

begin
    byte state [4, Nb]

    state = in

    AddRoundKey ( state, w [Nr* Nb, (Nr+1) * Nb-1])

    for round = Nr-1 step -1 downto 1
        InvShiftRows (state)
        InvSubBytes (state)
        AddRoundKey (state, w [ round * Nb, (round+1)* Nb-1])
        InvMixColumns (state)
    end for

    InvShiftRows (state)
    InvSubBytes (state)
    AddRoundKey (state, w [0, Nb-1])


    out = state
end
```

Fig. 4.3. Pseudo Code for the AES Inverse Cipher [http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf]

4.2.2 S-box

Here, a brief description of substitution by S-box is presented. The material is mainly borrowed from [http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf].

The substitution by S-box is a non-linear byte substitution that operates independently on each byte of the input matrix using a substitution table (S-box). This S-box, which is invertible, is constructed by composing two transformations:

1. Take the multiplicative inverse in the finite field $GF(2^8)$, the element {00} is mapped to itself.

2. Apply the following transformation:

$$b'_i = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i \tag{4.1}$$

for $0 \le i < 8$, where b_i is the i^{th} bit of the byte b, and c_i is the i^{th} bit of a byte c where the byte $c = \{63\}$ or $\{01100011\}$.

In matrix form, the transformation element of the S-box can be expressed as:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{4.2}$$

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Fig. 4.4. AES S-box: substitution values for the byte *xy* (in hexadecimal format)

[http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf]

The S-box values used for substitution are presented in Fig. 4.4 in hexadecimal format. For example, if input to S-box is $S = \{35\}$, then convert it to an 8-bit value {00100011}, where the higher nibble {0010} serves as index for row and lower nibble servers as index for column {0011}, then the substitution value would be determined by the intersection of the row with index ‘2’ and the column with index ‘3’ in Fig. 4. This would result in an output value $S' = \{26\}$. Rows and columns are numbered from 0 to 15.

4.2.3 Inverse S-box

Inverse S-box is the inverse of the S-box used during the encryption. The inverse S-box works in similar fashion as the S-box. The values for the inverse S-box are shown in Fig.4.5.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Fig. 4.5. AES Inverse S-box: substitution values for the byte *xy* (in hexadecimal format) [http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf].

4.3 HyChIES

HyChIES is a complete image encryption scheme. In this section Hybrid Chaotic Image Encryption Scheme (HyChIES) is presented in detail. HyChIES is a cryptosystem that employs multiple PLCMs, a generalized logistic map, an AES S-box and ciphertext feedback.

In the proposed scheme, each pixel value is substituted by the S-box. The effect of S-box and key is then diffused over all other pixels through a generalized logistic map. Three key streams, one each for R, G and B channels, named as K_R, K_G and K_B, are generated from three different PLCMs. The lengths of K_R, K_G and K_B are equal to 128 bytes each. The same K_R, K_G and K_B are reused for masking the full stream of the R, G and B channels. The extraction of K_R, K_G and K_B is rendered very difficult due to the S-box and chaotic diffusion process. After two iterations, the effect of the each pixel and key streams K_R, K_G and K_B is diffused rest of the pixels in the image. Due to the efficient diffusion process in HyChIES, it fulfills the avalanche property of ciphers, in that a change in any one pixel value leads to change in almost all other pixel values. In the proposed scheme, three rounds of enciphering is sufficient to cause considerable encryption.

HyChIES, as will be shown later, proves to be of very high security and, at the same time, has considerably less computational complexity compared to the other recently proposed chaotic encryption scheme, where huge numbers of iterations of the chaotic systems are used for the encryption of only one pixel. In what follows, we describe each and every step of the proposed image encryption scheme. However, before going into detailed discussion of HyChIES, we first introduce PLCM and generalized logistic map.

4.3.1 Piecewise Linear Chaotic Map (PLCM)

PLCM is a 1-D chaotic map that has been widely explored and used. As we have already mentioned, chaotic systems operate on real numbers. Generally, given a real interval, $X = [\alpha, \beta] \subset R$, a piecewise linear chaotic map $F : X \rightarrow X$ is a multi-segmented map such that for $i = 1$ to m, $C_i = F_i(x) = a_i x + b_i$. Any $\{C_i\}_{i=1}^{m}$ constitutes a partition of X, and $\bigcup_{i=1}^{m} C_i = X$. Also $C_i \cap C_j = \emptyset, \forall i \neq j$. Such a map has the following statistical properties:

- It is chaotic since its Lyapunov exponent is positive
- It is exact
- It has the property of mixing
- It is ergodic
- It has uniform invariant density function $f(x) = 1/(\beta - \alpha)$. The uniform invariant density function means that uniform density input will generate uniform density output, and the chaotic orbit from almost every initial condition will lead to the same uniform distribution $f(x) = 1/(\beta - \alpha)$.

4.3.2 Generalized Chaotic Logistic Map

Generalized logistic map is a discrete map, suitably modified from its continuous counterpart popularly known as logistic map. The modification is made so as to obtain integer values ranging from 0 to 255.

$$f(y_i) = \begin{cases} \left| \left\lfloor \dfrac{y_i(256 - y_i)}{64} \right\rfloor \right| & \tilde{y}_i < 256 \\ 255 & \tilde{y}_i = 256 \end{cases} \tag{4.3}$$

Here $f(y_i)$ is the output of logistic map, y_i is the input to the logistic map and $\tilde{y}_i = \left\lfloor \frac{y_i(256-y_i)}{64} \right\rfloor$.

4.3.3 Initialization of HyChIES

4.3.3.1 Mapping Two Dimensional R, G and B Channels into One Dimensional Stream

In this section we show how the two dimensional R, G and B matrices are mapped into 1-D stream. In order to map into 1-D stream, an invertible scanned pattern is used. The scan pattern used in the HyChIES is shown in Fig.4.6. It employs reading the pixel values column wise sequentially. This is an invertible process and hence the image can be reconstructed when deciphered into the same 1-D stream. The one dimensional streams are called as R^{1D}, G^{1D} and B^{1D}.

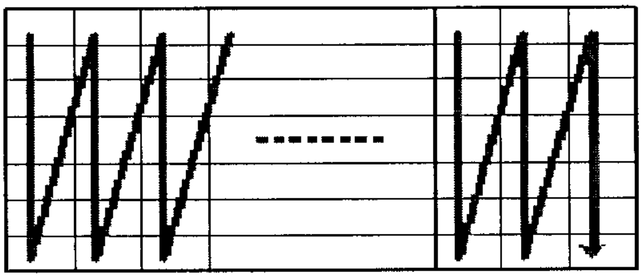


Fig. 4.6. Scan pattern for reading the R, G and B- Matrices

4.3.3.2 Extracting a Unique Initial Condition from External Secret Key

In this step, the initial condition, *IC*, for three chaotic key streams that are used to mask the three one-dimensional streams R^{1D}, G^{1D} and B^{1D} is derived. The initial condition is same for all the three PLCMs used for generating the streams, however, the control parameters for the PLCMs are different, and hence, the three chaotic key streams are also different. The initial condition, *IC*, is derived by using 16 PLCMs – each deriving its own initial condition, IC_i from the external secret key of 16 characters. The secret key in ASCII form is denoted by (4.4).

$$K = K_1 K_2 K_3 \cdots K_{16} \tag{4.4}$$

Here K_i denotes the 8-bit block of the secret key. Using K_i, the i^th initial condition IC_i is given by (4.5):

$$IC_i = {K_i}/{256} \tag{4.5}$$

Now, the initial condition, *IC*, for the three PLCMs is derived in the following fashion.

Let the PLCM be defined by (4.6).

$$CM(x,p) = \begin{cases} x/p & 0 \le x \le p \\ (x-p)/(1/2-p) & p \le x < 1/2 \\ CM(1-x,p) & 1/2 < x \le 1 \end{cases}, \tag{4.5}$$

where $0 < p < 1/2$, and x serves as the initial condition

Using IC_i to replace x and employing different control parameter, p_i for each of the 16 PLCMs, one can generate the initial condition, *IC*, as follows:

$$R = \sum_{i=1}^{16} CM_i^{K_i}(IC_i, p_i)$$
$$IC = R \bmod 1 \tag{4.7}$$

where R refers to an intermediate variable. K_i refers to the number of iteration of the i^{th} chaotic system with corresponding initial condition IC_i and control parameter p_i.

The values of 16-control parameters p_i that are used in carrying out the analysis are given in (4.8).

$$\begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \\ p_{13} & p_{14} & p_{15} & p_{16} \end{bmatrix} = \begin{bmatrix} 0.15 & 0.1535 & 0.1555 & 0.1666 \\ 0.27 & 0.2745 & 0.2755 & 0.2777 \\ 0.43 & 0.4344 & 0.444 & 0.4377 \\ 0.45 & 0.4566 & 0.4576 & 0.4578 \end{bmatrix} \tag{4.8}$$

4.3.3.3 Generating the Chaotic Key Stream for Masking the 1-D data Stream

HyChIES employs three PLCMs for the encryption of the R^{1D}, G^{1D} and B^{1D} matrices. The PLCM is one of the simplest chaotic maps. The three PLCMs are denoted as follows:

$$CM_1 = CM_1(IC, p_1), CM_2 = CM_2(IC, p_2), CM_3 = CM_3(IC, p_3) \tag{4.9}$$

where p_1, p_2 and p_3, three different control parameters and PLCMs utilizes the same initial condition derived from the external secret key. The PLCMs, like all other chaotic maps, are known to be highly sensitive to variations in the initial condition and control parameters. In our analysis, we have taken $p_1 = 0.15$, $p_2 = 0.27$ and $p_3 = 0.43$. The chaotic key stream from *m*-PLCMs is generated using (4.10).

$$K_k = \lfloor CM_k * (10^n) \rfloor (\bmod 256) \quad where \ k = 1,2,3 \tag{4.10}$$

Here K_k, the key stream from k^{th} PLCM is represented in its 8-bit value, and n is dependent on the implementation precision. In our analysis of HyChIES, we have taken $n = 7$ giving rise to a periodicity of the order of 10^7.

From hereon, K_1, K_2 and K_3 will be referred as K_R, K_G and K_B, respectively. In each round of HyChIES, different chaotic stream is generated for masking the streams of R^{1D}, G^{1D} and B^{1D} matrices.

4.3.4 HyChIES Encryption Algorithm

After generating key stream and mapping the two dimensional R, G and B channels into their corresponding R^{1D}, G^{1D} and B^{1D} matrices, the encryption of an image is performed by 1) substituting the pixel values from R^{1D}, G^{1D} and B^{1D} channels using an S-box, and, 2) masking substituted streams with the chaotic key stream and mixing them with previous ciphered pixel (ciphertext feedback). It completes one round of encryption. A few rounds of encryption usually suffice for an efficient cipher design.

4.3.4.1 Substitution by S-box

Substitution by S-box is a nonlinear transformation step that plays a very significant role in fighting against the differential and linear cryptanalysis. HyChIES uses S-box of AES which has maximum differential and linear probabilities of 2^{-6} and 2^{-3} respectively [Daemen & Rijmen, 2002]. In HyChIES, each pixel value is substituted by the S-box. The output pixel value from the S-box is XORed with a chaotic key stream from output of the generalized logistic map. The effect of the S-box is diffused over all other pixels by the generalized logistic map. Hence after two rounds, the effect of each pixel values is diffused overall rest of the pixels. The combination of S-box and chaotic mixing makes the overall encryption scheme secure. Details regarding the functionality and design philosophy of S-box can be found in [Daemen & Rijmen, 2002].

Let the pixel values in the R^{1D}, G^{1D} and B^{1D} matrices be denoted, respectively, by (4.11).

$$[x_1, x_2, x_3, \cdots\cdots, x_N],\ [y_1, y_2, y_3, \cdots\cdots, y_N],\ \text{and}[z_1, z_2, z_3, \cdots\cdots, z_N] \tag{4.11}$$

After substitution by S-box shown in Fig.4.3, the corresponding pixel values are represented by (4.12).

$$[sx_1, sx_2, sx_3, \cdots\cdots, sx_N],\ [sy_1, sy_2, sy_3, \cdots\cdots, sy_N],\ \text{and}[sz_1, sz_2, sz_3, \cdots\cdots, sz_N] \tag{4.12}$$

4.3.4.2 Masking and Mixing: based on generalized logistic map and ciphertext feedback

Mixing causes diffusion of data. The reason for introducing diffusion in an encryption algorithm is that it can significantly change the statistical properties of the plain-image by spreading the influence of each bit of the plain-image all over the cipher-image. For a secure encryption scheme, a diffusion step is mandatory; otherwise the opponent can break the cryptosystem by comparing a pair of plain-text and cipher-text and discovering some useful information from it.

In HyChIES, a generalized logistic map is used for diffusion purpose that is given in (4.3). The generalized logistic map operates on previous ciphered pixel (ciphertext feedback) and the corresponding key stream values K_R, K_G and K_B. The output of the generalized logistic map is XORed with the output of the pixel value substituted by the S-box. Hence, by using the generalized chaotic logistic map, the effect of each pixel and key is diffused over rest of the pixels in an image. The process of the key addition and chaotic mixing is illustrated in (4.13).

$$\begin{gathered}
\underbrace{sx_1 \oplus f(K_R(1) \oplus x'_N)}_{x'_1}, \underbrace{sx_2 \oplus f(K_R(2) \oplus sx_1)}_{x'_2}, \underbrace{sx_3 \oplus f(K_R(3) \oplus x'_2)}_{x'_3}, \cdots\cdots \\
, \underbrace{sx_N \oplus f(K_R((k \bmod L)+1) \oplus x'_{N-1})}_{x'_N}, \\
\underbrace{sy_1 \oplus f(K_R(1) \oplus y'_N)}_{y'_1}, \underbrace{sy_2 \oplus f(K_G(2) \oplus sy_1)}_{y'_2}, \underbrace{sy_3 \oplus f(K_G(3) \oplus y'_2)}_{y'_3}, \cdots\cdots \\
, \underbrace{sy_N \oplus f(K_G((k \bmod L)+1) \oplus y'_{N-1})}_{y'_N}, \\
\text{and} \\
\underbrace{sz_1 \oplus f(K_B(1) \oplus z'_N)}_{z'_1}, \underbrace{sz_2 \oplus f(K_G(2) \oplus sz_1)}_{z'_2}, \underbrace{sz_3 \oplus f(K_G(3) \oplus z'_2)}_{z'_3}, \cdots\cdots \qquad (4.13) \\
, \underbrace{sz_N \oplus f(K_B((k \bmod L)+1) \oplus z'_{N-1})}_{z'_N}
\end{gathered}$$

where x'_i, y'_i, and z'_i, represent the updated pixel value. $L = 128$ is the length of K_R, K_G and K_B and k serves as an index for acquiring a byte from K_R, K_G and K_B and the value of k is between 0 and $L-1$.

From (4.13), it is evident that $x'_2 = sx_2 \oplus f(K_R(2) \oplus sx_1)$ whereas $x'_1 = sx_1 \oplus f(K_R(1) \oplus x'_N)$. In other words, while encrypting the second pixel, the ciphered value for pixel 2, x'_2, is obtained first and x'_1 - the ciphered value for pixel 1 is obtained last. So, in order to encipher the second pixel, sx_2 - the substituted value for pixel 2, is XORed with the output of the generalized logistic map – the input to the logistic map being key stream $K_R(2)$ and sx_1 - the first substituted pixel value. Similarly, the encryption for the first pixel is obtained by XORing sx_1 - the substituted value, with the output of the generalized logistic map – the input to the logistic map being key stream $K_R(1)$ and x'_N - the last ciphered pixel value.

It is also evident from (4.13) that the same key stream is repeatedly used for masking the next 128 pixels after substituting them by the S-box. The reuse of the chaotic key stream reduces the computational load of HyChIES, without reducing the security level, as the extraction of the key streams K_R, K_G and K_B is made infeasible by the inclusion of the S-box and chaotic diffusion process.

4.3.5 HyChIES Decryption Algorithm

In HyChIES, the decryption process consists of the similar steps. All steps employed in the encryption process are reversible.

4.3.5.1 Inverse of Masking and Mixing

Adding the same chaotic key stream becomes its own inverse as shown herein. In encryption all the chaotic key streams are extracted on the fly. However, during the decryption process, three key streams are extracted from each PLCM at the start. Let the three extracted key streams from each of the PLCMs be denoted as:

$$A = \left[K_R^1, K_R^2, \text{and } K_R^3\right],$$
$$B = \left[K_G^1, K_G^2, \text{and } K_G^3\right]$$

and

$$C = \left[K_B^1, K_G^2, \text{and } K_G^3\right] \quad (4.14)$$

where A, B and C represents the three extracted key streams from the 1st ,2nd and 3rd PLCM respectively, where each one is of the length 128 bytes.

During decryption process, key streams in A, B and C are used in the reverse order i.e. K_R^3 first, then K_R^2, followed by K_R^1. However, for the sake of convenience, we will still refer to the corresponding key streams as K_R, K_G and K_B.

The decryption is described by (4.15):

$$
\begin{aligned}
&\underbrace{x_1' \oplus f(K_R(1) \oplus x_N')}_{sx_1}, \underbrace{x_2' \oplus f(K_R(2) \oplus sx_1)}_{sx_2}, \underbrace{x_3' \oplus f(K_R(3) \oplus x_2')}_{sx_3}, \cdots\cdots \\
&, \underbrace{x_N' \oplus f(K_R((k \bmod L)+1) \oplus x_{N-1}')}_{sx_N}, \\
&\underbrace{y_1' \oplus f(K_R(1) \oplus y_N')}_{sy_1}, \underbrace{y_2' \oplus f(K_G(2) \oplus sy_1)}_{sy_2}, \underbrace{y_3' \oplus f(K_G(3) \oplus y_2')}_{sy_3}, \cdots\cdots \\
&, \underbrace{y_N' \oplus f(K_G((k \bmod L)+1) \oplus y_{N-1}')}_{sy_N}, \\
&\text{and} \\
&\underbrace{z_1' \oplus f(K_B(1) \oplus z_N')}_{sz_1}, \underbrace{z_2' \oplus f(K_G(2) \oplus sz_1)}_{sz_2}, \underbrace{z_3' \oplus f(K_G(3) \oplus z_2')}_{sz_3}, \cdots\cdots \\
&, \underbrace{z_N' \oplus f(K_B((k \bmod L)+1) \oplus z_{N-1}')}_{sz_N}
\end{aligned}
\tag{4.15}
$$

During encryption process, the pixel value is substituted first and then followed by XOR operation with the output of the generalized logistic map that operates on the previous pixel value and corresponding byte from the key streams. However, as evident from (4.15), during decryption, the ciphered value of the first pixel is XORed with the output of the generalized logistic map – the inputs to the logistic map being $K_R(1)$ and the ciphered value for the N^{th} pixel, x_N', and then inverse substituted by the inverse S-box. In other words, we first obtain $sx_1, sx_2 \cdots sx_N$ and then use inverse substitution to obtain $x_1, x_2 \cdots x_N$. However, the cipher having a ciphertext feedback step, the sx_1 is obtained first followed by sx_2 etc in that order as described in (4.15) and then inverse substituted.

4.3.5.2 Inverse of Substitution by S-box

This step is the inverse of substitution transformation performed by S-box. Here each input pixel value is substituted by inverse S-box shown in Fig. 4.4.

4.4 H-CES

In this section, a 128-bit block cipher similar to AES is presented. It is a variant of HyChIES, in that only the initial input data is read in blocks of 128 bit each. All other steps are almost same.

4.4.1 Initialization of H-CES

4.4.1.1 Extracting a Unique Initial Condition from External Secret Key

Here, a unique initial condition for the 16-PLCMs used in the generation of the chaotic key stream is extracted. The process of extracting the IC is similar to that of HyChIES, described in the section 4.3.1.2.

4.4.1.2 Generating Chaotic Key Stream

H-CES employs 16-PLCMs for masking the input plain block of 16 bytes. 16-PLCMs are denoted as follows:

$$CM_1 = CM_1(IC, p_1), \cdots\cdots, CM_{16} = CM_{16}(IC, p_{16}) \tag{4.16}$$

where p_i is the control parameter for the i^{th} PLCM. The control parameters used for 16-PLCMs in our analysis are given in (4.8). The chaotic key stream from m-PLCMs is generated by (4.17).

$$\phi_i(k) = \lfloor CM_i * (10^7) \rfloor (\mathrm{mod}\, 256) \quad where\ i = 1,2,3,\cdots,16 \tag{4.17}$$

Here $\phi_i(k)$ represented in 8 bits, is the key stream from i^{th} PLCM.

The generated 128-bit chaotic key stream denoted by ϕ, is represented in matrix form in (4.18).

$$\phi(k) = [\phi_1(k) \quad \phi_2(k) \quad \cdots \quad \phi_{16}(k)] \tag{4.18}$$

4.4.2 H-CES cipher

In this section, the steps used in the H-CES cipher are described. Let the input to CES cipher be denoted as:

$$x = [x_1, x_2, x_3, \cdots\cdots, x_{16}] \tag{4.19}$$

where x_i represents an 8-bit block in the input 128 bit plain block.

4.4.2.1 Substitution by S-box

In this step each byte in the 128 bit input plain block is substituted by S-box, similar to AES. The output of this transformation is denoted as:

$$sx = [sx_1, sx_2, sx_3, \cdots\cdots, sx_{16}] \tag{4.20}$$

4.4.2.2 Masking and Mixing: based on generalized logistic map and ciphertext feedback

In this step the effect of the each active S-box is diffused over all other input bytes through generalized chaotic logistic map. This works in a similar fashion to HyChIES, except that it operates on 128-bit input plain block. The process of Masking and Mixing Transformation is denoted as:

$$\underbrace{sx_1 \oplus f(\phi(1) \oplus x'_{16})}_{x'_1}, \underbrace{sx_2 \oplus f(\phi(2) \oplus sx_1)}_{x'_2}, \underbrace{sx_3 \oplus f(\phi(3) \oplus x'_2)}_{x'_3}, \cdots\cdots$$
$$, \underbrace{sx_N \oplus f(\phi(16)) \oplus x'_{15})}_{x'_{16}}, \tag{4.21}$$

where x'_i, represents the i^{th} ciphered value. From (4.21), it can be seen that the each byte depends on all the previous outputs. After two iterations the effect of each bit is diffused over all other bits in the input matrix.

4.4.3 H-CES Inverse Cipher

In this section, the steps regarding H-CES inverse cipher are described. Let the input to H-CES inverse cipher be denoted as:

$$x'_i = \lfloor x'_1, x'_2, x'_3, \cdots\cdots, x'_{16} \rfloor \tag{4.22}$$

where x'_i represents 8-bit block in the input 128 bit cipher block.

4.4.3.1 Inverse of Masking and Mixing

During encryption, the chaotic key streams are extracted on the fly, however during the decryption process, key streams of 128 bits from m-PLCMs are extracted at the start. Let the key streams for n-rounds be denoted as:

$$A = \lfloor \phi^1 \quad \phi^2 \quad \cdots \quad \phi^n \rfloor \tag{4.23}$$

where A, represents $n -$ 128 bit extracted key streams from 16-PLCM.

During the decryption process, key streams in A, are used in the reverse order i.e. ϕ^n first, then ϕ^{n-1} and ϕ^1 is used in the last round of decryption. During encryption process, each byte is substituted first and then followed by XOR operation with the output of the generalized logistic map that operates on the previous ciphered byte and corresponding byte from the key streams. However during decryption, each byte in the input matrix is XORed with the output of the generalized logistic map and then inverse substituted by the inverse S-box. The inverse of masking and mixing transformation is denoted in (4.24).

$$\underbrace{x_1' \oplus f(\phi^n(1) \oplus x_N')}_{sx_1}, \underbrace{x_2' \oplus f(\phi^n(2) \oplus sx_1)}_{sx_2}, \underbrace{x_3' \oplus f(\phi^n(3) \oplus x_2')}_{sx_3}, \cdots\cdots$$

$$, \underbrace{x_N' \oplus f(\phi^n(16) + 1) \oplus x_{N-1}')}_{sx_{16}}, \tag{4.24}$$

4.4.3.2 Inverse of Substitution by S-box

This step is the inverse of the byte substitution by S-box. Here, the inverse S-box is applied to each byte of the input matrix. The inverse S-box is shown in Fig. 4.4.

4.5 Summary

In this chapter, we described each and every step of the proposed HyChIES and H-CES in detail. HyChIES operates on full image while H-CES operates on 128-bit block like AES. The heart of both cryptosystems is same – consisting of an AES S-box, generalized logistic map and ciphertext feedback. The key stream for both cryptosystems is extracted from a stream cipher that employs 16-PLCMs. The analysis results of both schemes will be presented in the next chapter.

5

CHAPTER

Analysis of the Proposed Encryption Schemes

5.1 Introduction

This chapter details out the analysis of the both ciphers described in the last chapter. Analysis of the HyChIES is presented first, followed by the security analysis of the proposed H-CES.

5.2 Analysis of HyChIES

In this section the proposed scheme are analyzed in detail from security and computational aspects. A good encryption procedure should be robust against all kinds of cryptanalytic, statistical and brute-force attacks. A detailed security analysis of the proposed image encryption scheme is presented by doing statistical analysis, sensitivity analysis with respect to the key and plaintext etc. in order to prove that the proposed cryptosystem is secure against the most common attacks.

5.2.1 Histogram Analysis

An image-histogram illustrates how pixels in an image are distributed by plotting the frequency of pixels at each color intensity level. The histograms of several encrypted as well as its original colored images are analyzed that have widely different content and the proposed scheme is seen to effectively randomize the plain images. Histograms of a few images obtained from [http://sipi.usc.edu/database/] are shown in Figs. 5.1, 5.2, and 5.3, after encrypting by HyChIES. The histogram suggests a fairly uniform probability density function of the ciphered images. The normalized mean squared error (NMSE), a measure of the deviation from uniform probability distribution, is calculated for each histogram using (5.1).

$$NMSE = \frac{1}{N}\sum_{k}\left(\frac{X_k - \overline{X}}{\overline{X}}\right)^2 \tag{5.1}$$

where N is the number of bins, equal to 256. X_k is the frequency of occurrence of each bin. $\overline{X}$ is the mean frequency (ideal frequency) of each bin.

NMSE corresponding to Figs. 5.1, 5.2 and 5.3 are 0.0013, 0.0017 and 0.0012, respectively.

Consider a plain image 4.1.03.tiff obtained from [http://sipi.usc.edu/database/]. The histograms of the R, G and B channels of a plain image [4.1.03.tiff] are displayed in Figs. 5.4, 5.6 and 5.8 respectively. While the histograms of the R, B and G channels of the corresponding encrypted image are shown in Figs. 5.5, 5.7 and 5.9 respectively. It is evident from displayed figures that the histograms of the encrypted image are fairly uniform and significantly different from the respective histograms of the corresponding original image. The NMSE indicating deviation from the uniform probability distribution for Figs. 5.5, 5.7 and 5.9 are 0.0041, 0.0038 and 0.0035, respectively. Given these proofs, it is very difficult to employ any statistical attack on the proposed image encryption procedure.

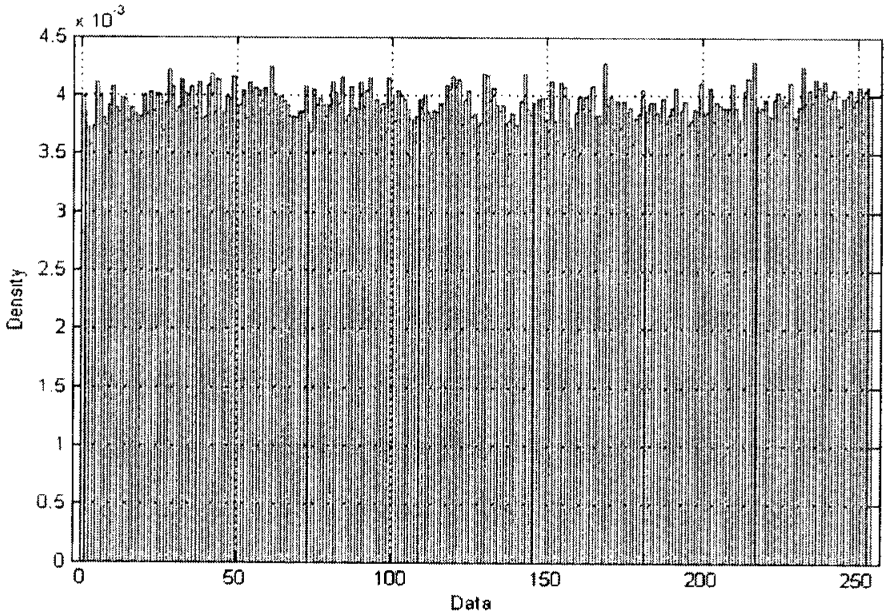


Fig. 5.1. Histogram of the ciphered image 4.1.01.tiff [http://sipi.usc.edu/database/]

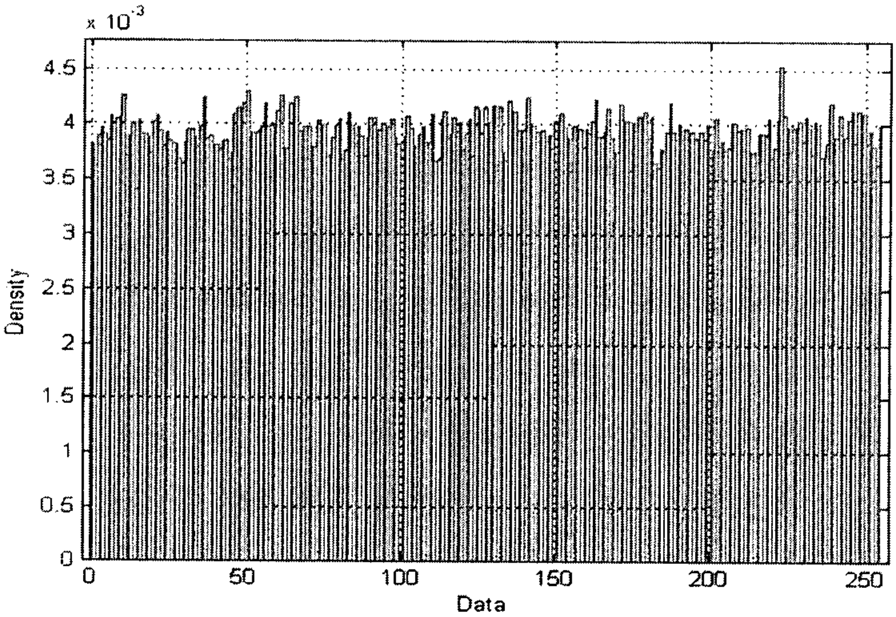


Fig. 5.2. Histogram of the ciphered image 4.1.02.tiff [http://sipi.usc.edu/database/]

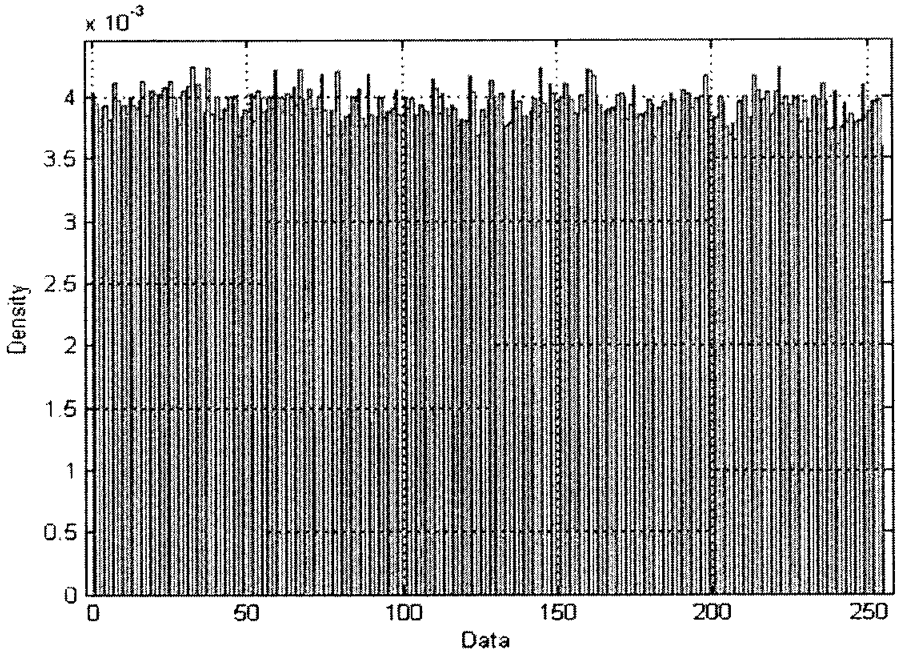


Fig. 5.3. Histogram of the ciphered image 4.1.03.tiff [http://sipi.usc.edu/database/]

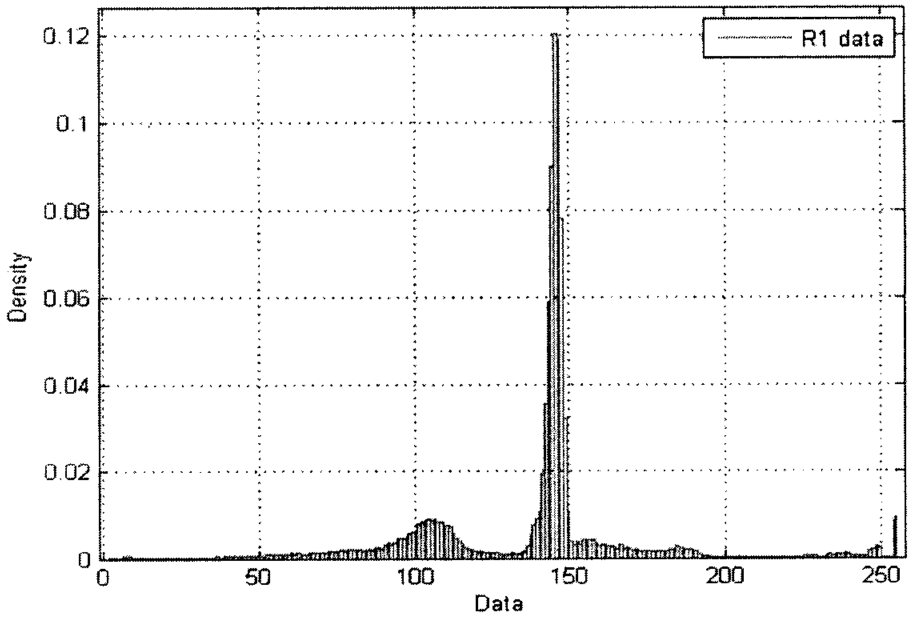


Fig. 5.4. Histogram of the R-channel of the plain image 4.1.03.tiff [http://sipi.usc.edu/database/]

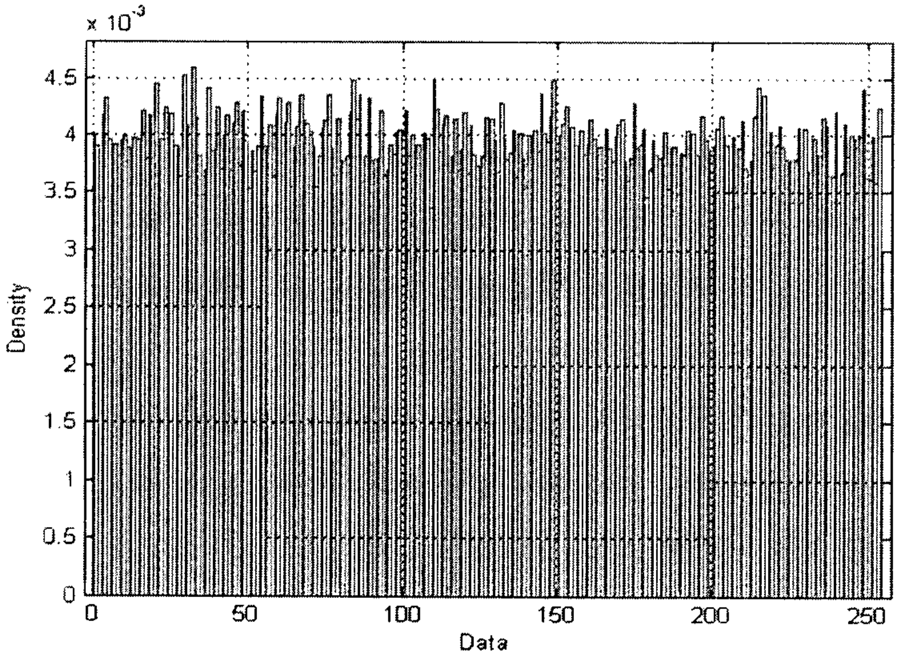


Fig. 5.5. Histogram of the R-channel of the corresponding ciphered image 4.1.03.tiff [http://sipi.usc.edu/database/]

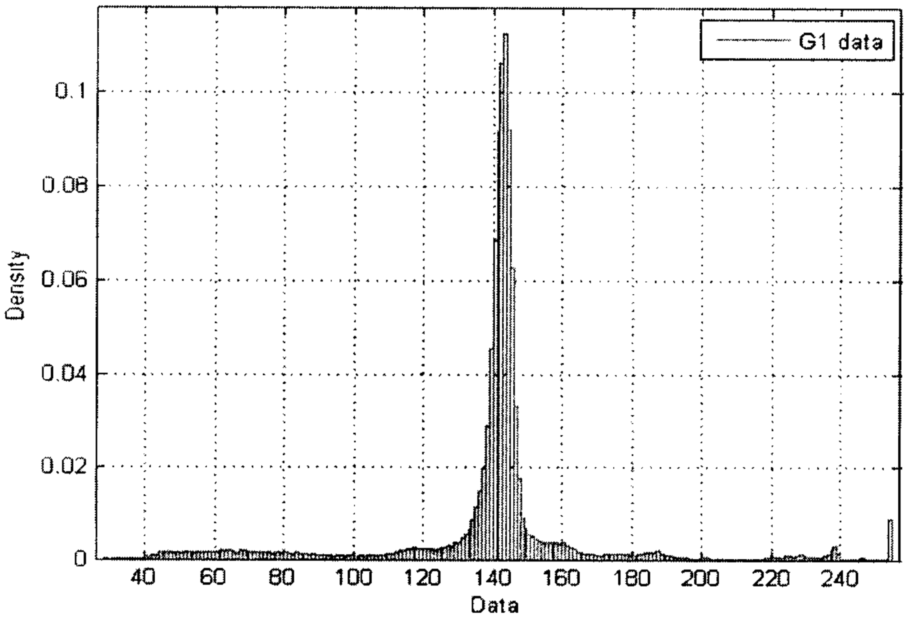


Fig. 5.6. Histogram of the G-channel of the plain image 4.1.03.tiff [http://sipi.usc.edu/database/]

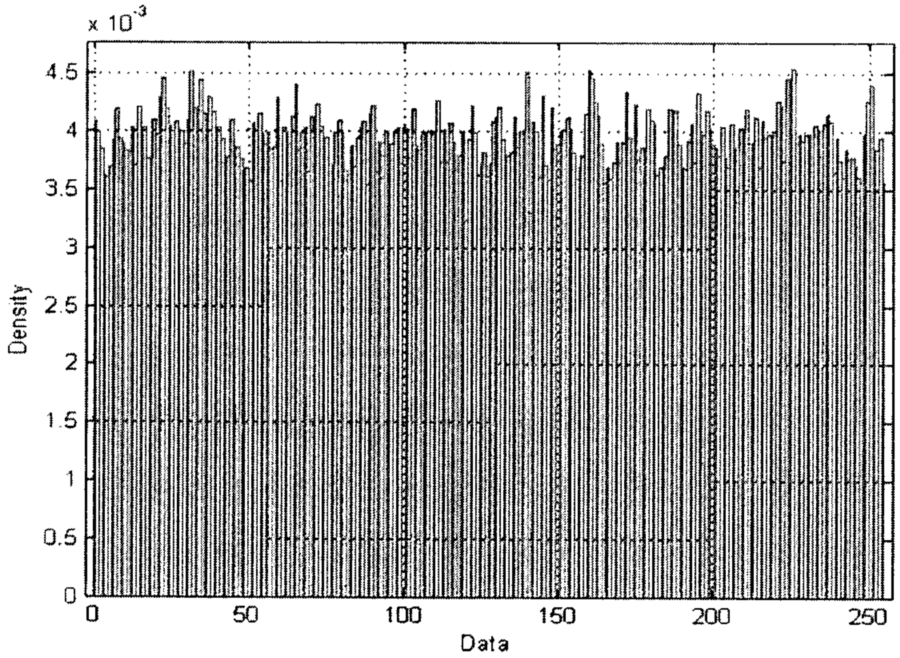


Fig. 5.7. Histogram of the G-channel of the corresponding ciphered image 4.1.03.tiff [http://sipi.usc.edu/database/]

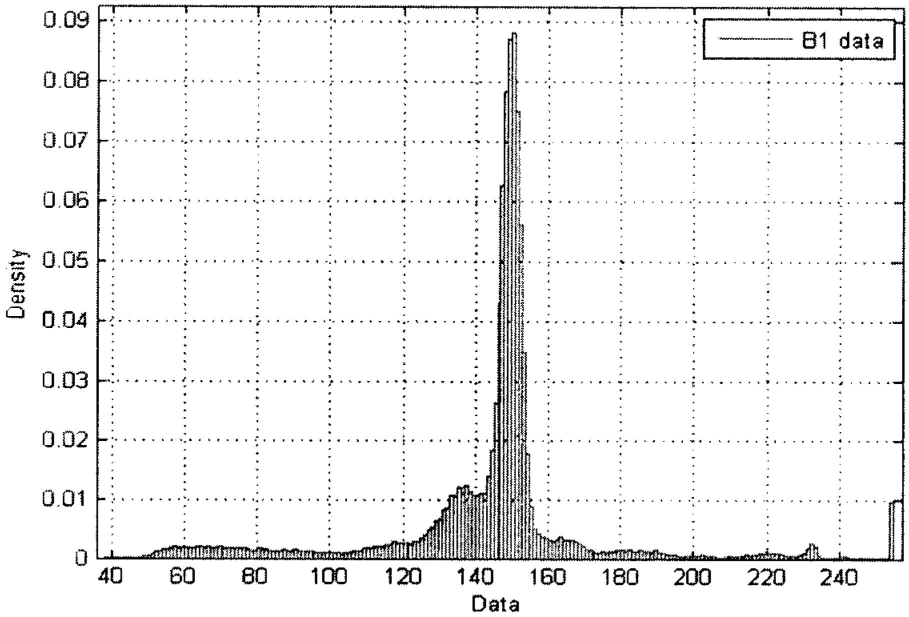


Fig. 5.8. Histogram of the B-channel of the plain image 4.1.03.tiff [http://sipi.usc.edu/database/]

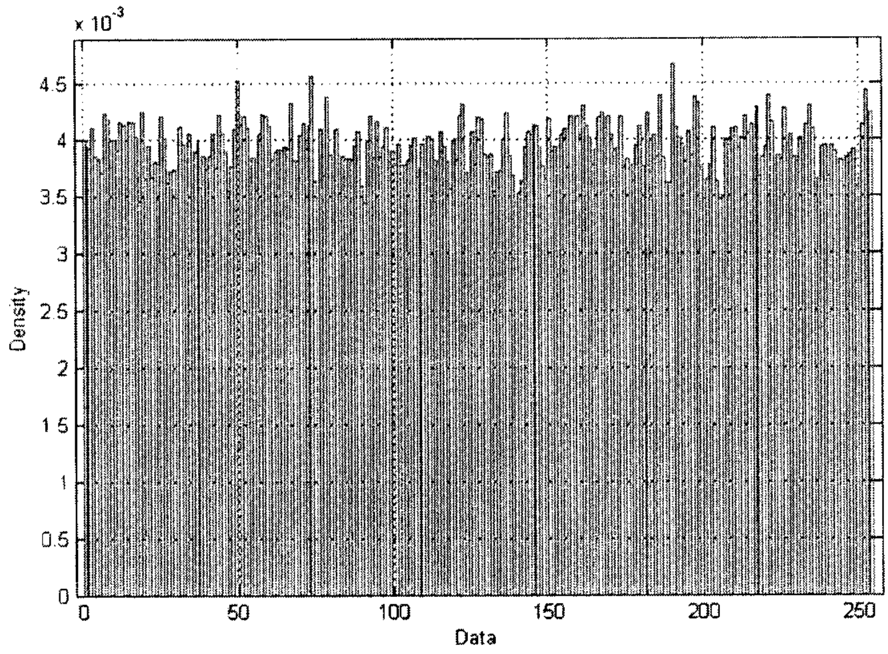


Fig. 5.9. Histogram of the B-channel of the corresponding ciphered image 4.1.03.tiff [http://sipi.usc.edu/database/]

5.2.2 Correlation Coefficient Analysis

In this section the images are analyzed on the basis of the correlation between the adjacent pixels. In this test, a frame consisting of 2500 gray level values are taken from the R, G and B channels of the plain and ciphered images. Figs. 5.10, 5.12 and 5.14 show scatter plots for a frame taken from the R, G and B channels of a plain image [4.1.01.tiff] obtained from [http://sipi.usc.edu/database/] respectively. In Figs. 5.11, 5.13 and 5.15, the scatter plots for a frame taken from the R, G and B channels of the corresponding cipher image are shown respectively. In order to calculate the correlation coefficient between the adjacent pixels, the following definition (5.2) of the correlation is used.

$$r = \frac{\sum_m \sum_n \left(A_{mn} - \bar{A}\right)\left(B_{mn} - \bar{B}\right)}{\sqrt{\left(\sum_m \sum_n \left(A_{mn} - \bar{A}\right)^2\right)\left(\sum_m \sum_n \left(B_{mn} - \bar{B}\right)^2\right)}} \qquad (5.2)$$

where $\bar{A}$ and $\bar{B}$ represents the corresponding mean values. Here r represents the normalized correlation between image A_{mn} & B_{mn} pixel by pixel.

Fig. 5.10 shows the adjacent pixel correlation for a frame from the R channel of a plain image, 4.1.01.tiff. The figure has a band of points along the diagonal indicating strong correlation between adjacent pixels. The normalized correlation, r, for this case is 0.9021. On the other hand, Fig. 5.11 shows the correlation between the adjacent pixels of the same frame after encrypting it by HyChIES. It is seen that pixel values are almost uniformly distributed in the interval 0 & 255 and the correlation coefficient is 0.0021, which is quite negligible.

Fig. 5.12, 5.13, 5.14, 5.15 shows the adjacent pixel correlation for a frame of G and B channels before and after encryption. It is seen that after encrypting by HyChIES, the correlation coefficient is quite negligible.

The correlation between the various plain images obtained from a freely available online data base in [http://sipi.usc.edu/database/] and their corresponding cipher images are extensively studies using the proposed encryption scheme, where the all of the images are encrypted using the secret key '*abcdefghijklmnop*'. The correlation coefficients are shown in Table 5.1. It is evident that the average correlation coefficient is very small which implies that no correlation exists between original and its corresponding cipher image. In all of the above cases the correlation coefficient are computed using (5.2).

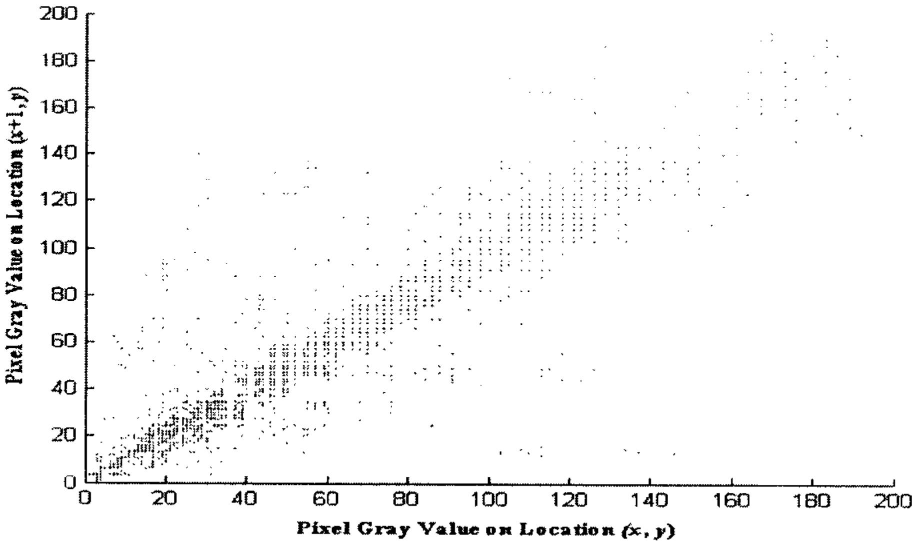


Fig. 5.10. Horizontal correlation for plain R-channel: Correlation of two horizontally adjacent pixels for a frame taken from R channel of the plain image 4.1.01.tiff [http://sipi.usc.edu/database/], where the correlation coefficient = 0.9021

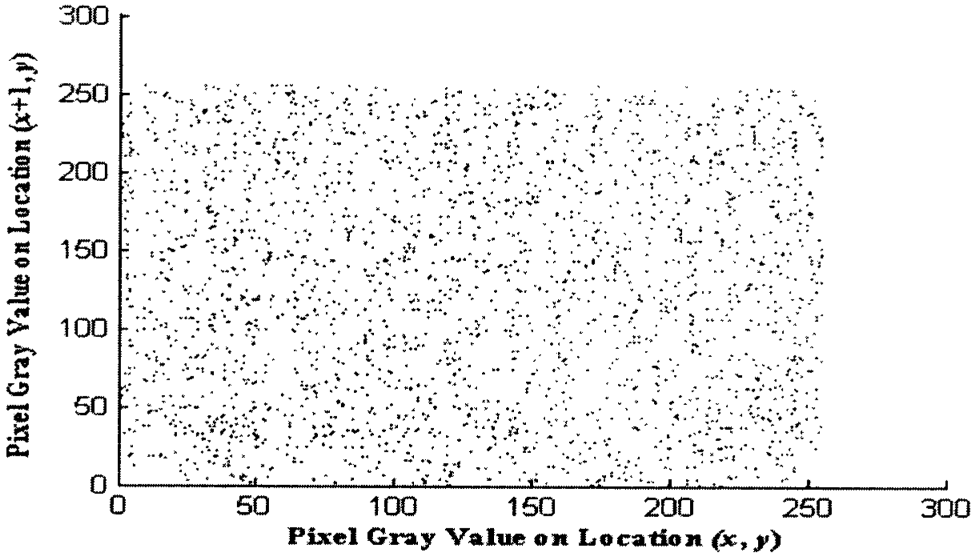


Fig. 5.11. Horizontal correlation for ciphered R-channel: Correlation of two horizontally adjacent pixels for a frame taken from R channels of the corresponding cipher image of 4.1.01.tiff [http://sipi.usc.edu/database/], where the correlation coefficient = 0.0021

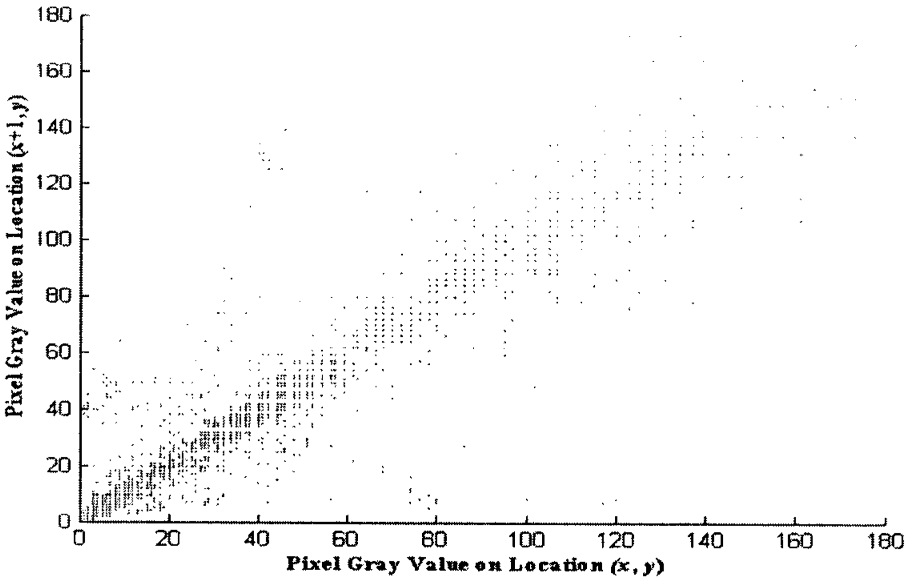


Fig. 5.12. Horizontal correlation for plain G-channel: Correlation of two horizontally adjacent pixels for a frame taken from G channel of the plain image 4.1.01.tiff [http://sipi.usc.edu/database/], where the correlation coefficient=0.9233

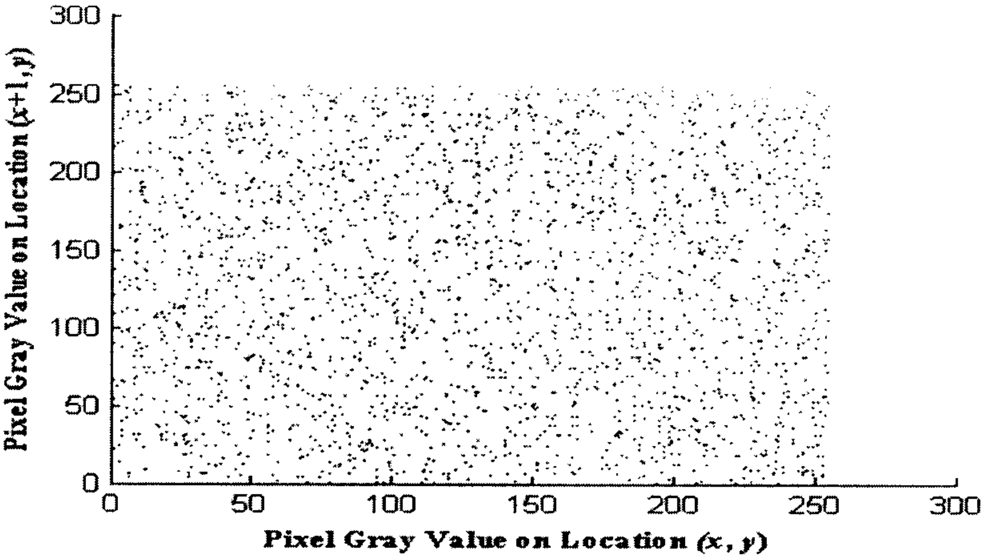


Fig. 5.13. Horizontal correlation for ciphered G-channel: Correlation of two horizontally adjacent pixels for a frame taken from G channel of the corresponding cipher image of 4.1.01.tiff [http://sipi.usc.edu/database/], where the correlation coefficient= -0.0045

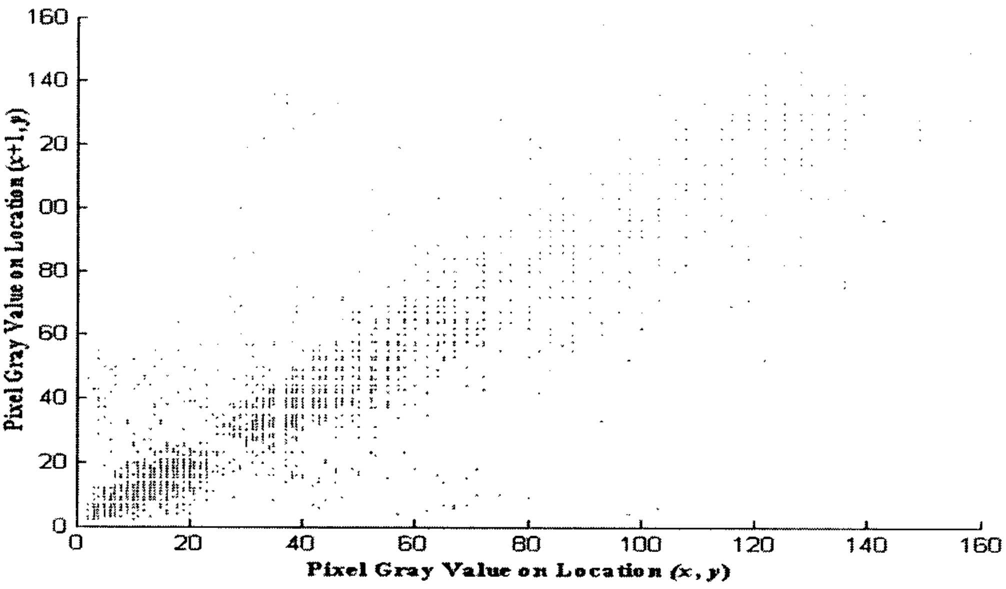


Fig. 5.14. Horizontal correlation for plain B-channel: Correlation of two horizontally adjacent pixels for a frame taken from B channel of the plain image 4.1.01.tiff [http://sipi.usc.edu/database/], where the correlation coefficient = 0.9059

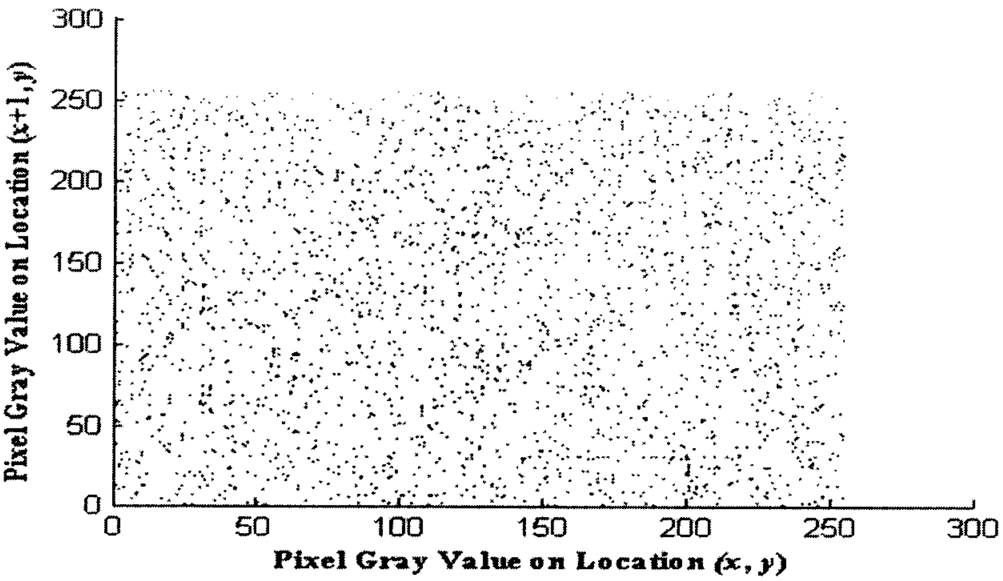


Fig. 5.15. Horizontal correlation for ciphered B-channel: Correlation of two horizontally adjacent pixels for a frame taken from B channel of the corresponding cipher image of 4.1.01.tiff [http://sipi.usc.edu/database/], where the correlation coefficient = 0.0041.

Table 5.1. Correlation coefficient analysis: Correlation coefficient between the image and corresponding cipher image for a number of images obtained from [http://sipi.usc.edu/database/]. The encryption has been done using the secret key '*abcdefghijklmnop*'.

File Name	File Description	Type	Size	Correlation coefficient
4.1.01	Girl	Color	256×256	0.0061
4.1.02	Couple	Color	256×256	0.00094804
4.1.03	Girl	Color	256×256	0.000070541
4.1.04	House	Color	256×256	0.00050198
4.1.05	Tree	Color	256×256	0.00016302
4.1.06	Jelly beans	Color	256×256	0.0015
4.1.07	Jelly beans	Color	256×256	-0.00095852
4.2.01	Splash	Color	512×512	0.00029972
4.2.02	Girl(Tiffany)	Color	512×512	-0.0037
4.2.03	Baboon	Color	512×512	0.0004335
4.2.04	Girl(Lena)	Color	512×512	-0.00034129
4.2.05	Airplane(F-16)	Color	512×512	0.00096007
4.2.06	Sailboat on Lake	Color	512×512	-0.0012
4.2.07	Peppers	Color	512×512	-0.00018774

5.2.3 Key Sensitivity Analysis

The cipher image obtained by HyChIES is extremely sensitive to the change in secret key. Ideally a single bit of difference between encrypting and decrypting key should make it unable to decrypt the ciphered image. In other words, it can be said that a single bit of change in an encrypting key should result in totally different encrypted image. For testing the key sensitivity of the proposed image encryption scheme, the following experimental analysis has been performed:

- An original image in Fig. 5.16 is encrypted using the secret key '*abcdefghijklmnop*', and shown in Fig. 5.17.

- Tried to decrypt the encrypted image Fig. 5.17 is decrypted with slightly wrong key '*abcdefghijklmnoo*' that differs from the correct secret key in the least significant character. The decrypted image with the slightly wrong key and its histogram are shown in Figs. 5.18 and 5.19 respectively,

- The same ciphered image of Fig. 5.17 is also decrypted with another slightly wrong key '*abcdefghijklmmop*', which also differs from the correct key with only one character. The decrypted image and its histogram are shown in Figs. 5.19 and 5.20 respectively.

It is evident that it is not possible to decrypt an image with a slightly different key and that HyChIES is highly key sensitive.

Fig. 5.16. Plain image 4.1.01.tiff [http://sipi.usc.edu/database/]

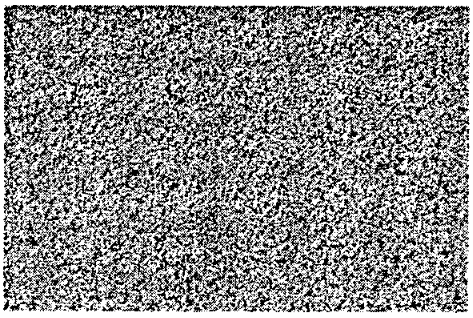


Fig. 5.17. Ciphered image of 4.1.01.tiff: Ciphered image of the corresponding plain image 4.1.01.tiff [http://sipi.usc.edu/database/] encrypted using the secret key '*abcdefghijklmnop*'

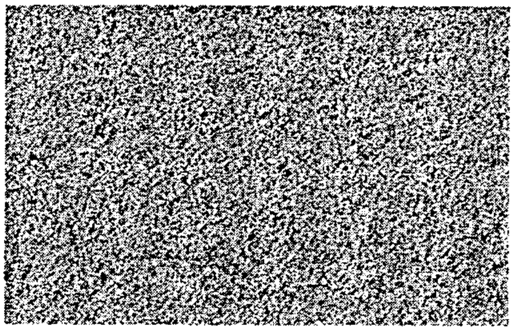


Fig. 5.18. Decrypted image with wrong key '*abcdefghijklmnoo*'

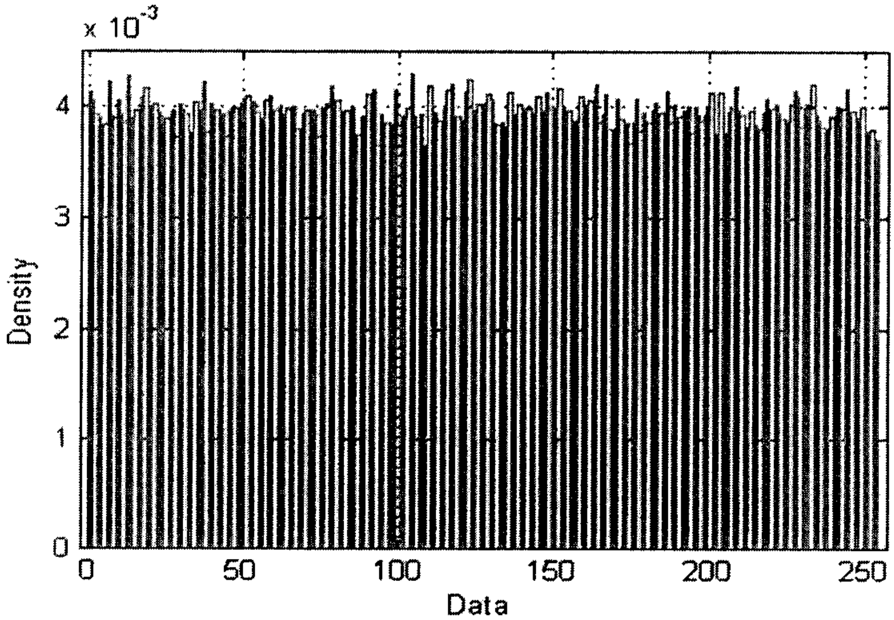


Fig. 5.19. Histogram of image decrypted with wrong key '*abcdefghijklmnoo*'

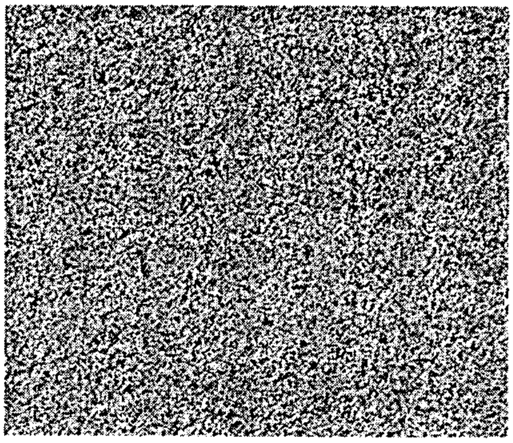


Fig. 5.20. Decrypted image with wrong key '*abcdefghijklmmop*'

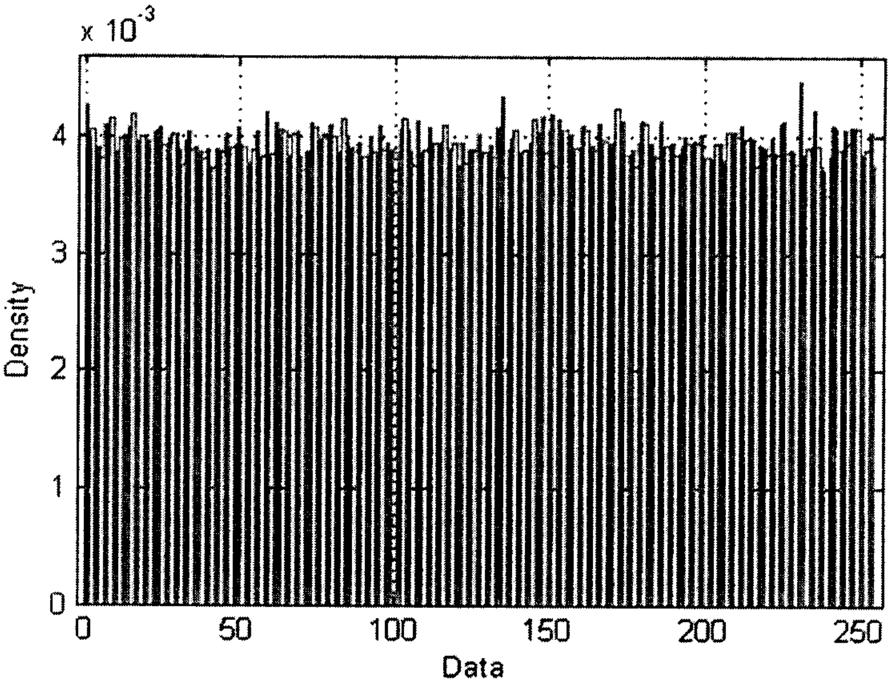


Fig. 5.21. Histogram of image decrypted with wrong key '*abcdefghijklmmop*'

5.2.4 Sensitivity to Plain Image

In this case, the sensitivity of the proposed scheme to the change in plain image is investigated. In general, the attacker may make a slight change (e.g., modify only one pixel) to the plain image, and then observe the change in the result. In this way, he may be able to find out a meaningful relationship between the plain image and the cipher image. Usually these types of approaches are used in the differential attacks. Hence, if one minor change in the plain-image can cause a significant change in the cipher-image, then this differential attack would become very inefficient and practically useless.

In order to test the influence of one-pixel change on the whole image encrypted by HyChIES, a measure known as number of pixels change rate (NPCR) is calculated. Let us denote two cipher-images, whose corresponding plain-images have only one-pixel difference, be denoted by C_1 and C_2, respectively. Label the grey-scale values of the pixels at grid (i, j) of C_1 and C_2, by $C_1(i, j)$ and $C_2(i, j)$, respectively. Define a bipolar array, D, with the same size as image C_1 and C_2. Then, $D(i, j)$ is computed by the following sub-program.

$$\begin{aligned} &\text{if } C_1(i,j) = C_2(i,j), \text{ then} \\ &\quad D(i,j) = 1; \\ &\quad \text{otherwise} \\ &\quad D(i,j) = 0. \end{aligned} \tag{5.3}$$

The NPCR is defined as:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{5.4}$$

where W and H are the width and height of C_1 and C_2. The NPCR measures the percentage of pixels different between the two images.

The NPCR for a few images is shown in table 5.2. It is evident that a single change in pixel will result in completely different ciphered images. The NPCR in table 5.2 is for three rounds of HyChIES. Generally, with the increase of ciphering rounds, the influence of one-pixel change is increased. However, this is at the expense of processing speed.

Table 5.2 NPCR for a number of images: Plain images differ in only single pixel position. The encryption has been done using the secret key '*abcdefghijklmnop*'

File Name	File Description	Size	NPCR
4.1.01	Girl	256×256	20.122 %
4.1.02	Couple	256×256	23%
4.1.03	Girl	256×256	21.3%
4.1.04	House	256×256	25.55%
4.1.05	Tree	256×256	20.98%
4.1.06	Jelly beans	256×256	23.2%
4.1.07	Jelly beans	256×256	22.88%

5.2.5 Differential Attack

In this section, it is shown that HyChIES is secure against our proposed attack on Pareek et al. chaotic image encryption scheme, which has been presented in chapter 3 of this thesis.

Assume that an $M \times N$ plain-image f_{Barbra} [4.1.01.tiff] and its corresponding cipher-image f'_{Barbra} have been known to an attacker. Then a mask image f_m is constructed by simply XORing the two ciphered images f'_{Barbra} and f'_{Girl}, pixel by pixel:

$$f_m(i,j) = f'_{\text{Barbra}}(i,j) \oplus f'_{\text{Girl}}(i,j), \qquad (5.5)$$

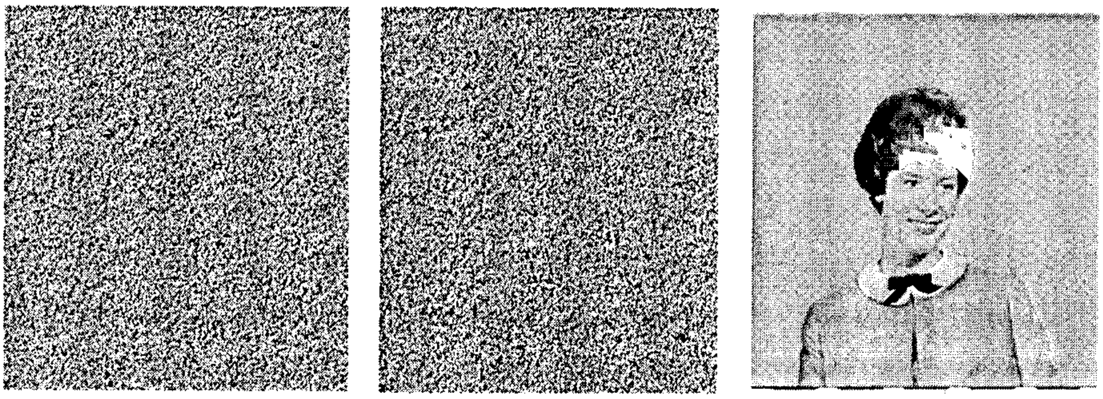
where $0 \le i \le M-1$ and $0 \le i \le N-1$.

With the mask image f_m, the attacker tries to recover the plain-image by XORing the mask image and known plain-image f_{Barbra} pixel by pixel:

$$f_R(i,j) = f_{\text{Barbra}}(i,j) \oplus f_m(i,j), \tag{5.6}$$

where f_R is the recovered image.

Figs. 5.22, 5.23 and 5.24 show the results of the above analysis. Decrypted image using the known plain image f_{Barbra} , and mask image f_m is shown in Fig. 5.22 (b), while the corresponding original image is shown in Fig. 5.22(c). It is evident that it is not possible to recover the original image by using f_m (consisting of differentials) and one known plain image f_{Barbra}. The histograms of mask image f_m (formed from two ciphered images) and mask image from their corresponding plain images are shown in Fig.5.23 and F.24 respectively. It is evident that the correlation between the two is negligible and that the histogram of f_m is very uniform.



a) f_m: Mask image **b)** Decrypted image using f_m **c)** Original image f_{Girl}

Fig. 5.22. Differential attack: Using f_m and one known plain image f_{Barbra} [4.1.01.tiff].

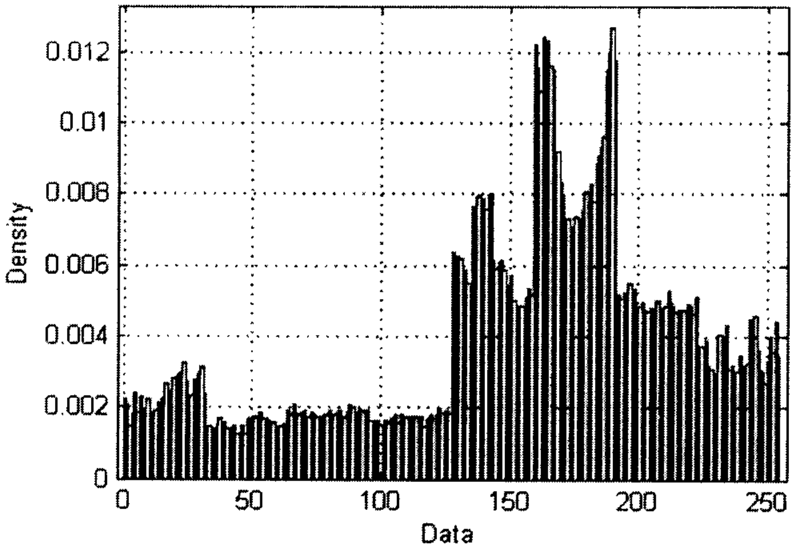


Fig. 5.23. Histogram of mask image obtained from plain images: f_{Barbra} [4.1.01.tiff] and f_{Girl} [4.1.03.tiff].

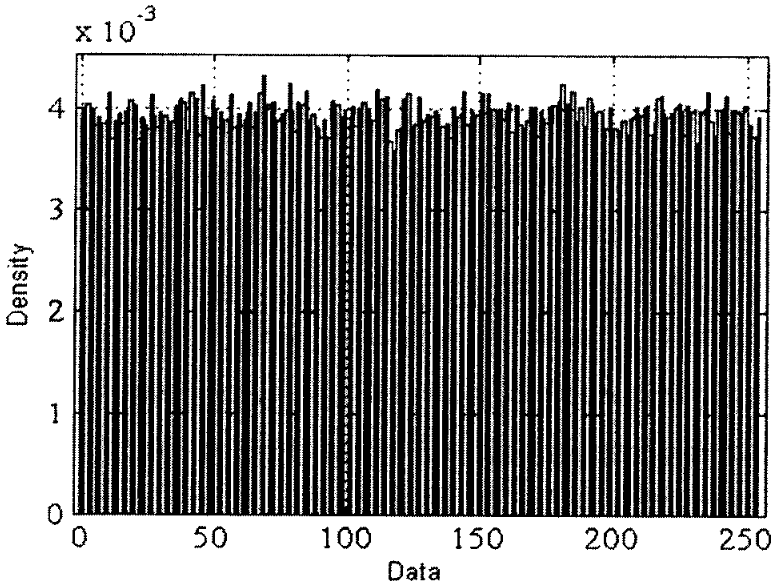


Fig. 5.24. Histogram of mask image f_m obtained from cipher images: f'_{Barbra} and f'_{Girl}.

5.2.6 Key Space Analysis

Since a cryptosystem cannot exist without a key, it must be clearly defined what parameters serve as the key. Once the key has been defined, it is equally important to study the key space in depth. The size of the key space is the number of encryption/decryption key pairs that are available in the cipher system.

In the proposed HyChIES and H-CES, a scheme similar to Pareek et al. and Fei et al. ciphers [Pareek et al., 2005] [Fei et al., 2005] has been used. It uses an external secret key of 128-bit that is further divided into 8-bit blocks and mapped to an initial condition of a chaotic scheme. Since the scheme employs external secret key that is mapped to an IC, two factors are very important to be considered in the mapping process.

- Positional significance of each character in the external secret key
- Key space size after the mapping process

5.2.6.1 Positional Significance of Each Character in the External Secret Key

As mentioned before, in Pareek et al. and Fei et al. ciphers [Pareek et al., 2005] [Fei et al., 2005], external secret key is employed that is mapped to an IC. In both ciphers, it is claimed that the key space size is 2^{128} based on the 128 bit external secret key. However during the mapping process the positional significance of the each character in the secret key is not retained and hence different combinations of the secret key consisting of the same characters resulted in the same IC. The key space size is obviously less than 2^{128}.

However during the mapping process of the external secret key to some intermediate variables in HyChIES and H-CES, the positional significance of the each character is retained. The sub-algorithm used for mapping the external secret key to IC, is contained in the equations (5.7), (5.8), (5.9) and (5.10), which are given as:

$$K = K_1 K_2 K_3 \cdots K_{16} \tag{5.7}$$

$$IC_i = K_i/256 \tag{5.8}$$

$$R = \sum_{i=1}^{16} CM_i^{K_i}(IC_i, p_i) \tag{5.9}$$

$$IC = R \bmod 1 \tag{5.10}$$

Equation (5.7) represents external secret key that consists of 16-characters. In equation (5.8), each character is uniformly mapped to a real number in the interval $(0,1)$, that is represented by IC_i which serves as the initial condition for i^{th} PLCM with control parameter p_i. In (5.9), i^{th} PLCM with initial condition IC_i and control parameter p_i, is iterated K_i times to derive 16 random numbers that are summed together resulting in the intermediate variable, R.

During the mapping process to the intermediate variable R, the positional significance of each character in the secret key can be seen to be retained. From equation (5.8), it is evident that the i^{th} character in the secret key serves as initial condition for the i^{th} PLCM with a control parameter p_i, and PLCM is iterated K_i times, where K_i is the ASCII value of the corresponding i^{th} character. Hence different combinations of the secret key consisting of the same characters will result in different random numbers that are summed together, resulting in different values for the intermediate variable, R. Equation (5.10) converts R to a value between 0 and 1.

Table 5.3 shows that different combinations of the external secret key that consists of the same characters result in different values for the intermediate variable, R, and, hence , different values for IC. During the analysis the following control parameters are used for the 16 PLCMs that are represented in the matrix form in (5.11).

$$\begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \\ p_{13} & p_{14} & p_{15} & p_{16} \end{bmatrix} = \begin{bmatrix} 0.15 & 0.1535 & 0.1555 & 0.1666 \\ 0.27 & 0.2745 & 0.2755 & 0.2777 \\ 0.43 & 0.4344 & 0.444 & 0.4377 \\ 0.45 & 0.4566 & 0.4576 & 0.4578 \end{bmatrix} \tag{5.11}$$

5.2.6.2 Key Space Size after the Mapping Process

In this section, the size of the key space is analyzed after mapping the external secret keys to *ICs*. The analysis is carried out to demonstrate that 128-bit external secret key which has 2^{128} possible combinations relate to almost 2^{128} different initial conditions, *ICs*. In the analysis, ***rand* ( )** function is used in MATLAB® to generate 100000 random values for the each character K_i in the external secret key $K = [K_1 \; K_2 \cdots K_{16}]$, where $0 \leq K_i \leq 255$.

The histogram for one of the i^{th} character generated by Distribution Fitting Tool in MATLAB® is shown in Fig. 5.25. The histogram of IC after the mapping process is shown in Fig. 5.26. From the Fig. 5.26, it is evident that the distribution is quite uniform and the mapping process results in almost the equal number of *ICs* as the number of external secret keys. Hence the mapping process does not change the key space size of the HyChIES and results in equivalent number of secret keys.

Table 5.3. ICs for typical external secret keys

External Secret Key-*K*	R	IC
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	8.1608	0.1608
{‘b’ ‘a’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	8.2293	0.2293
{‘a’ ‘b’ ‘d’ ‘c’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	8.9418	0.9418
{‘a’ ‘b’ ‘c’ ‘d’ ‘f’ ‘e’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	8.3958	0.3958
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘h’ ‘g’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	6.8415	0.8415
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘j’ ‘i’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	8.2245	0.2245
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘l’ ‘k’ ‘m’ ‘n’ ‘o’ ‘p’}	9.3886	0.3886
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘n’ ‘m’ ‘o’ ‘p’}	8.718	0.718
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘p’ ‘o’}	8.1739	0.1739
{‘p’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘a’}	8.0573	0.0573
{‘a’ ‘o’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘b’ ‘p’}	8.296	0.296
{‘a’ ‘b’ ‘n’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘c’ ‘o’ ‘p’}	8.4682	0.4682
{‘a’ ‘b’ ‘c’ ‘m’ ‘e’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘l’ ‘d’ ‘n’ ‘o’ ‘p’}	8.8678	0.8678
{‘a’ ‘b’ ‘c’ ‘d’ ‘l’ ‘f’ ‘g’ ‘h’ ‘i’ ‘j’ ‘k’ ‘e’ ‘m’ ‘n’ ‘o’ ‘p’}	9.2833	0.2833
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘k’ ‘g’ ‘h’ ‘i’ ‘j’ ‘f’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	8.5431	0.5431
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘j’ ‘h’ ‘i’ ‘g’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	7.7326	0.7326
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘i’ ‘h’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’}	7.7317	0.7317
{‘o’ ‘p’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘I’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘a’ ‘b’}	7.2468	0.2468
{‘i’ ‘j’ ‘k’ ‘l’ ‘m’ ‘n’ ‘o’ ‘p’ ‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’}	6.7497	0.7497
{‘e’ ‘f’ ‘g’ ‘h’ ‘m’ ‘n’ ‘o’ ‘p’ ‘a’ ‘b’ ‘c’ ‘d’ ‘I’ ‘j’ ‘k’ ‘l’}	8.2779	0.2779
{‘a’ ‘b’ ‘c’ ‘d’ ‘I’ ‘j’ ‘k’ ‘l’ ‘e’ ‘f’ ‘g’ ‘h’ ‘m’ ‘n’ ‘o’ ‘p’}	8.68	0.68
{‘m’ ‘n’ ‘o’ ‘p’ ‘i’ ‘j’ ‘k’ ‘l’ ‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’}	9.0107	0.0107
{‘a’ ‘b’ ‘c’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘m’ ‘n’ ‘o’ ‘p’ ‘I’ ‘j’ ‘k’ ‘l’}	9.7047	0.7047
{‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘a’ ‘b’ ‘c’ ‘m’ ‘n’ ‘o’ ‘p’ ‘i’ ‘j’ ‘k’ ‘l’}	9.2388	0.2388
{‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘n’ ‘o’ ‘p’ ‘i’ ‘j’ ‘k’ ‘a’ ‘b’ ‘c’ ‘m’ ‘l’}	6.6594	0.6594
{‘j’ ‘k’ ‘a’ ‘b’ ‘c’ ‘m’ ‘l’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘n’ ‘o’ ‘p’ ’i’}	8.7049	0.7049
{‘m’ ‘l’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘n’ ‘o’ ‘j’ ‘k’ ‘a’ ‘b’ ‘c’ ‘p’ ‘i’}	6.0957	0.0957
{‘m’ ‘l’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘n’ ‘o’ ‘j’ ‘I’ ‘a’ ‘b’ ‘c’ ‘p’ ‘k’}	6.132	0.132
{‘m’ ‘l’ ‘d’ ‘e’ ‘f’ ‘g’ ‘h’ ‘n’ ‘k’ ‘a’ ‘b’ ‘c’ ‘p’ ‘I’ ‘o’ ‘j’}	7.4701	0.4701
{‘m’ ‘l’ ‘h’ ‘n’ ‘o’ ‘j’ ‘k’ ‘a’ ‘b’ ‘c’ ‘p’ ‘I’ ‘d’ ‘e’ ‘f’ ‘g’}	8.5193	0.5193

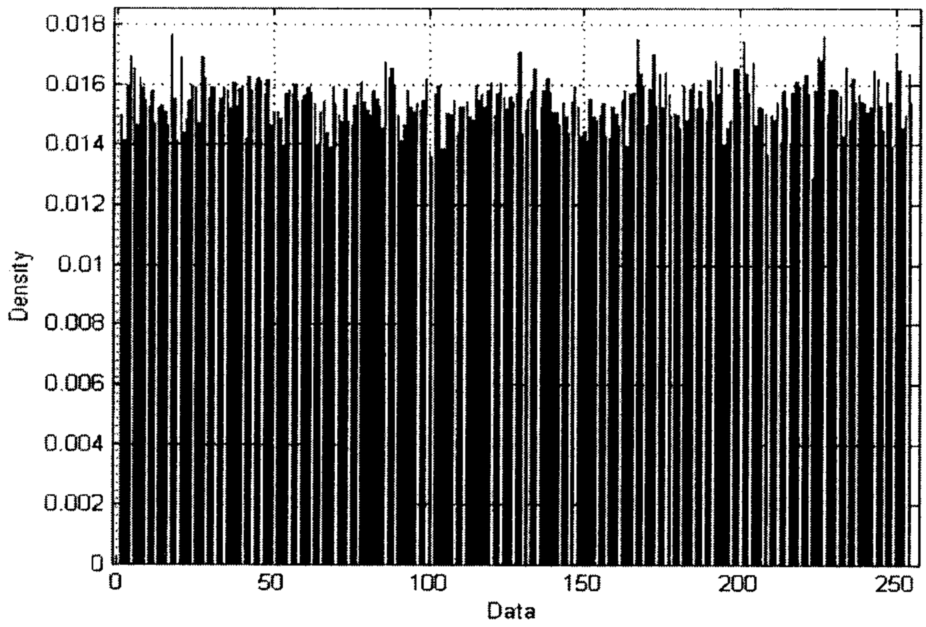


Fig. 5.25. 10000-bar histogram of K_i that consists of 100000 values

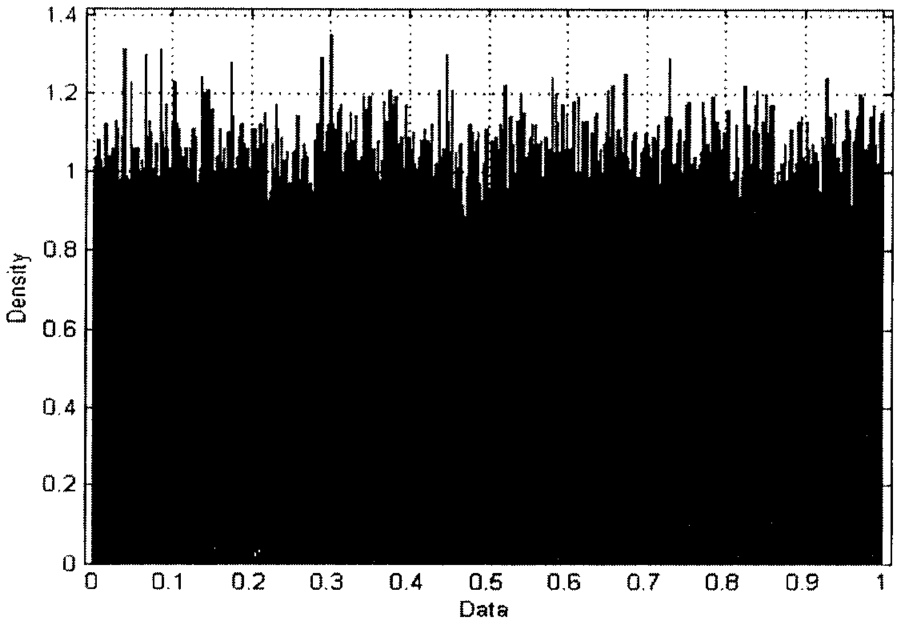


Fig. 5.26. 10000-bar histogram of *IC* after mapping 100000 external keys

5.2.7 Computational Complexity

The proposed HyChIES is of very low computational complexity compared to other recently proposed chaotic encryption schemes, where large number of iterations of the chaotic systems are performed for the encryption of one pixel [Pareek et al., 2006] [Fei et al., 2005] [Chen et al., 2004]. Along with the low computational complexity, HyChIES fulfills all the security requirements, which are demonstrated in the section 5.1.

In Table 5.4, the computational load of the recently proposed chaotic ciphers for the encryption of equivalent 8-bit is shown. In case of [Pareek et al., 2006], we assumed the number of iterations to be equal to 90, while in computing Fei et al.'s cipher, [Fei et al., 2005], which iterates the employed 3-D chaotic systems between 5 and 29 times, the computational load includes the load due to iterations of chaotic systems by 4th order Runge-Kutta method. In calculating the computational complexity of Fei et al.'s cipher, the number of iterations is averaged to 15.

Table 5.4. Computational complexity of HyChIES: Computations for encryption of equivalent one byte required by considered ciphers

Operations	1[1]	2[2]	3[3]	HyChIES (3 Rounds)
XORing	—	90	6	6
Multiplication	183.33	60	36	6
Shifting	—	—	—	
Substitution	—	30	—	3
Addition	108	—	21	
Subtraction	60	—	3	3
Modulo	3	*a*	12	—

[a]. May or may not be performed.

---

[1] [Fei et al., 2005]
[2] [Pareek et al., 2006]
[3] [Chen et al., 2004]

5.3 Analysis of H-CES

In this section, we analyze, the AES like 128 bit block cipher, H-CES and present the results of this analysis. First, H-CES is proved to be secure against the differential cryptanalysis and then some properties of H-CES are demonstrated.

The differential cryptanalysis exploits the propagation of difference (non-uniform mapping) from the round input to the round output. The probability with which a round input is mapped to a round output is referred to as differential probability. In general, the larger the differential probabilities of the active S-box, the larger the characteristic probability for the complete cipher. Also, fewer the active S-boxes are, the larger the characteristic probability. The complexity of differential cryptanalysis (number of chosen plaintext pairs required to mount the attack) required to break a cipher is inversely proportional to the differential characteristic probability of a cipher. Hence, it is considered vital to calculate the differential characteristic probability of a cipher in order to prove that it is secure against the differential cryptanalysis.

5.3.1 Differential and Linear Probabilities

Cryptographic security of a block-encryption algorithm can be checked only by means of proving its resistance to various kinds of known attacks. For example, one should prove (or check carefully) the resistance to differential and linear attacks. Providing the proof for resistance against differential and linear cryptanalysis methods is considered vital for proving the security of block ciphers like AES, HCES etc. But, one should keep in mind that provable security against one or two important attacks *does not imply* that the cipher is secure: other attacks may exist. On the other hand, provable security against certain attacks is certainly a first step in the right direction.

5.3.2 Differential Cryptanalysis

Differential cryptanalysis method was first published by Biham and Shamir in 1991 [Biham & Shamir, 1991]. It is a chosen plaintext (difference) attack in which a large number of plaintext-ciphertext pairs are used to determine the value of key bits. It basically exploits the mapping properties of the differences within the cryptosystem (S-box), since not every input difference can be mapped to an output difference. Statistical key information is deduced from ciphertext blocks obtained by encrypting pairs of plaintext blocks with a specific bitwise difference A' under the target key. The work factor of the attack, a measure of the amount of work needed to suggest the correct key, depends critically on the largest probability, $\max.\text{Prob}(B'/A')$, with B' being a difference at some fixed intermediate stage of the block cipher.

The input difference of the S-box for a round does not depend on the round key due to the fact that: $B' = (B \oplus K) \oplus (B^* \oplus K) = B \oplus B^*$, where B, B^* are inputs to S-box and B' is their corresponding difference.

5.3.3 Linear Cryptanalysis

Linear cryptanalysis is a known plaintext attack in which large numbers of plaintext-ciphertext pairs are used to determine the value of the key bits. Unlike differential cryptanalysis, no condition is imposed on the plaintext and its corresponding ciphertexts.

5.3.4 Differential and Linear Probabilities of an S-Box

Differential probability of a given S-box is a measure for differential uniformity. Let S be an S-box with m input and m output bits, then the differential probability of S is defined as:

$$DP^{S}(\Delta x \to \Delta y) = \left(\frac{\#\{x \in X | S(x) \oplus S(x \oplus \Delta x) = \Delta y\}}{2^{m}} \right) \tag{5.12}$$

where X is the set of all possible input values and 2^{m} is the number of its elements. Actually, DP^{S} is the probability of having output difference Δy, when the input difference is Δx.

Linear probability of a given S-box is defined as:

$$LP^{S}(\Gamma y \to \Gamma x) = \left[2.\left(\frac{\#\{x \mid x \bullet \Gamma x = S(x) \bullet \Gamma y\}}{2^{m}} \right) - 1 \right]^{2} \tag{5.13}$$

where Γx and Γy are input and output masks respectively, X the set of all possible inputs and 2^{m} the number of its elements. The linear probability is square of the imbalance of the event: the parity of the input bits selected by the mask Γx is equal to the parity of the output bits selected by the mask Γy. Decreasing LP^{S} implies increasing the complexity of the linear cryptanalysis attack.

5.3.5 Maximum Differential and Linear Probabilities of S-box

The maximum differential and linear probabilities of S-box are defined as:

$$DP^{S}_{\max} = \max_{\Delta x \neq 0, \Delta y} DP^{S}(\Delta x \to \Delta y)$$

and

$$LP^{S}_{\max} = \max_{\Gamma x,\, \Gamma y \neq 0} LP^{S}(\Gamma x \to \Gamma y) \tag{5.14}$$

respectively. The maximum linear probability of AES *S-box* is at most $q = 2^{-3}$, while the difference propagation probability is at most $p = 2^{-6}$.

5.3.6 Active S-box

An S-box is said to be active when its input/output is non zero. A differentially active S-box is defined as an S-box given a non-zero input difference and a linearly active S-box as an S-box given a nonzero output mask value.

5.3.7 Differential Cryptanalysis of H-CES

In this section, the effect of various blocks of H-CES on the difference propagation is analyzed. As it is already mentioned, the heart of H-CES cryptosystem, like before, consists of generalized logistic map, AES S-box and ciphertext feedback. Hence, in what follows, we analyze the mapping performed by generalized logistic map, and combination of generalized logistic map and ciphertext feedback, separately. After analyzing the mapping, difference propagation through the combination of generalized logistic map and ciphertext feedback is analyzed, followed by the analysis of the difference propagation through AES S-box and hybrid S-box separately, where hybrid S-box consists of the AE S-box, generalized logistic map and ciphertext feedback. After calculating the maximum differential probability for the hybrid S-box, differential characteristic probability for the two rounds of H-CES is computed.

5.3.7.1 Analysis of the Mapping performed by the Generalized Logistic Map

The generalized logistic map can be argued to be an S-box. However, unlike an S-box, the mapping is not one-to-one. In fact, all chaotic maps are not one-to-one. There are distinct elements of the set that are mapped to the same value. Thus, the cardinality of the set of all possible output values is less than 256. For example, the number of elements that are mapped to the value 255 is 17. This property implies that, when the input values are uniformly distributed, the output values are not uniformly distributed, i.e., the function “spoils” the input uniform distribution. Fig. 5.27 shows the mapping by the generalized logistic map for uniform input between 0 and 255. It is evident that, generally, most input values are mapped two to one, while in few cases more than two values are mapped to the same values. In particular 17 values are mapped to 255 (the frequency of occurrence of 255 is 17).

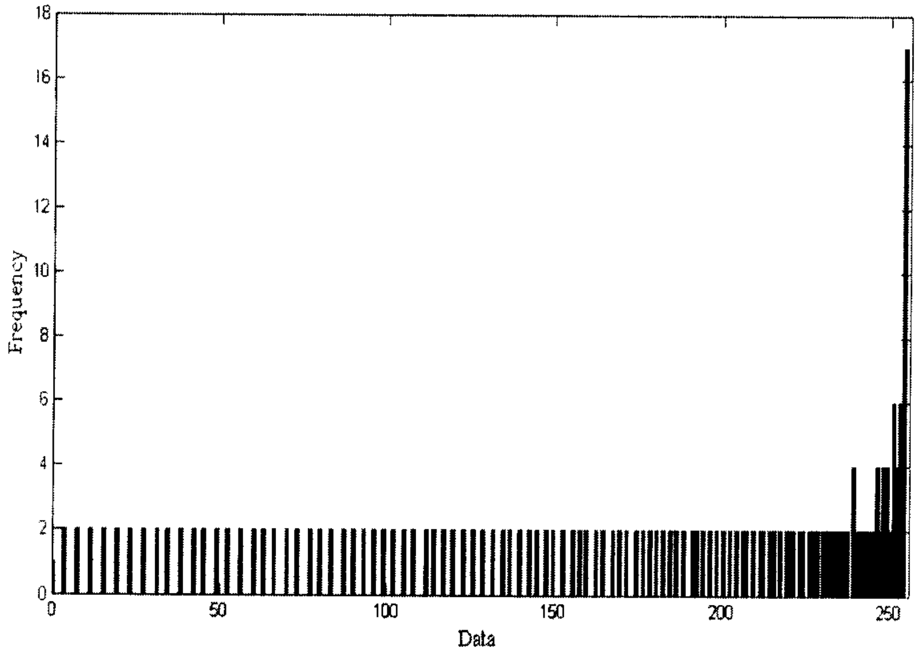


Fig. 5.27. Mapping performed by the generalized logistic map

5.3.7.2 Generalized Logistic Map and Ciphertext Feedback

The mapping by the generalized logistic map and ciphertext feedback is also analyzed. During the mapping, the following equation is considered:

$$y_i = x_i \oplus f(y_{i-1}) \qquad (5.15)$$

where x_i is the uniform input between 0 and 255, y_i is the current output value that is dependent on x_i and the previous output value y_{i-1}.

The result of the mapping by generalized logistic map and ciphertext feedback is shown in the Fig. 5.28. The result of the analysis is presented in the Table 5.5 that shows the number of input values and their corresponding number of output values after the mapping. It is evident that most of the values are now mapped to the distinct elements, while only two values occur with high frequency and the distribution is more uniform compared to the one performed by the generalized logistic map alone.

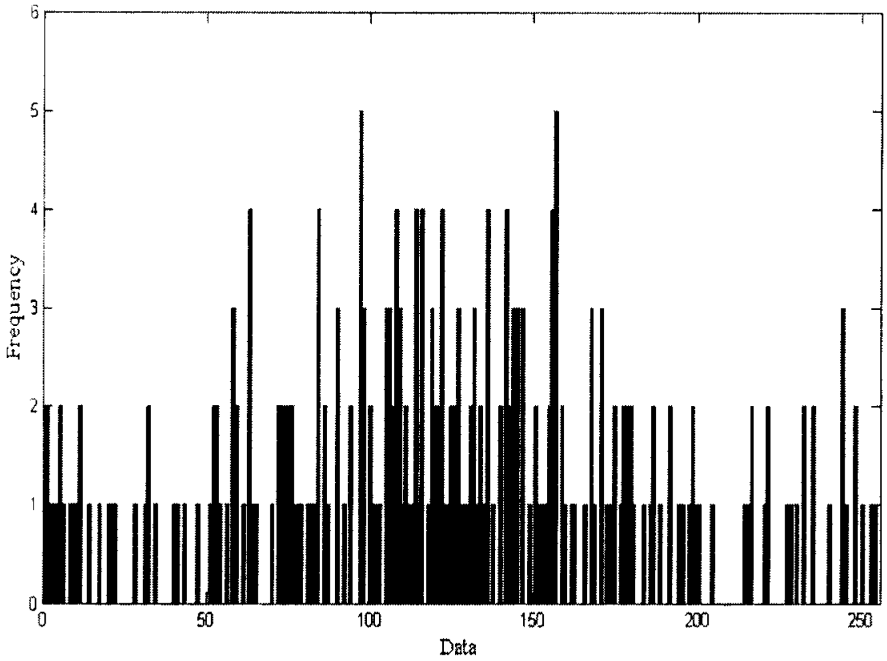


Fig. 5.28. Mapping performed by the generalized logistic map and ciphertext feedback

Table 5.5. Result of the analysis from Fig. 5.28

Number of inputs values	Corresponding Number Of output values	Frequency of occurrences
83	83	1
82	41	2
45	15	3
36	9	4
10	2	5

5.3.7.3 Differential Probability: Generalized Logistic Map and Ciphertext Feedback as an S-box

During the analysis, the logistic map and the ciphertext feedback were treated as an S-box and its effect on the difference propagation is analyzed. A difference distribution table is constructed, where rows represent Δx and columns represent the corresponding output difference Δy. The intersection of a particular row and column gives the probability of occurrence of corresponding output difference Δy, given Δx as the input difference. As it is not possible to report the full difference distribution table, a 3-D plot in MATLAB® is plotted using **mesh ( )** function, and shown in Fig. 5.29, where x-axis represents the input difference Δx, y-axis represents the corresponding output difference Δy, and z-axis represents the probability of occurrence for corresponding differentials $(\Delta x, \Delta y)$.

From Fig. 5.29, it is evident that some of the differentials occur with very high probabilities. The differential $(\Delta x, \Delta y) = (254, 254)$ occurs with the maximum probability of 45/256, while few of the differentials occur with the probability of 38/256. Hence the use of generalized logistic map and ciphertext feedback as an S-box is not advisable, as differentials are not effectively distributed. During the analysis, it is found that the situation is even worse if the generalized logistic map is used alone, without the ciphertext feedback.

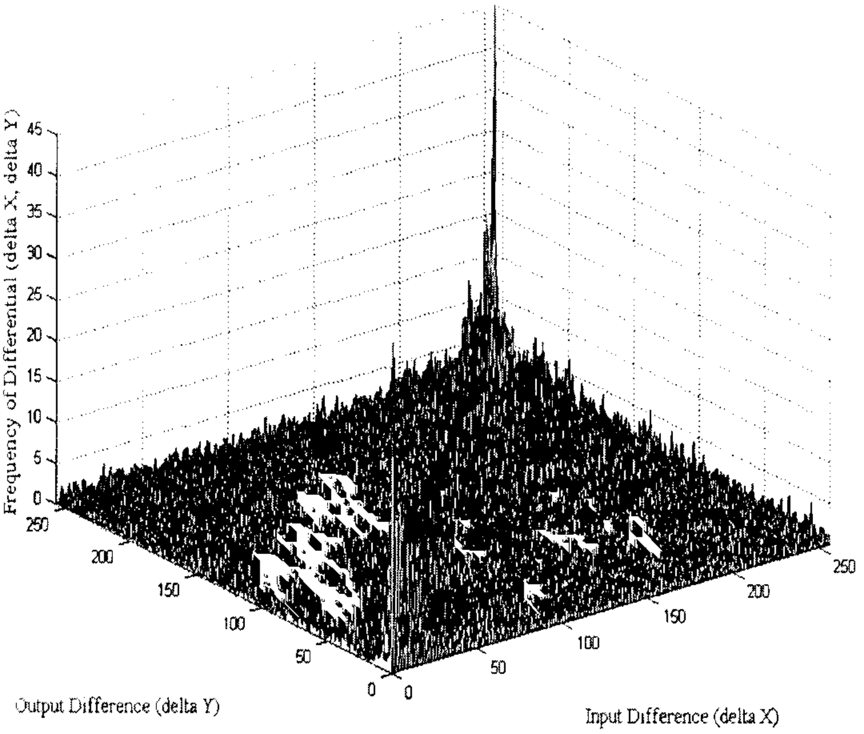


Fig. 5.29. Distribution of differentials for generalized logistic map and ciphertext feedback as S-box $(\Delta x, \Delta y)$.

5.3.7.4 Differential Probability of the Hybrid S-box: AES S-box, Generalized Logistic Map and Ciphertext Feedback

In this section, AES S-box and the generalized logistic map are viewed as a Hybrid S-box and will be represented by S'. The generalized logistic map operates on the previous ciphered byte. The structure of Hybrid S-box can be visualized as:

$$y_i = S(x_i) \oplus f(y_{i-1}) \qquad (5.16)$$

where y_i is the current ciphered byte and y_{i-1} is the previous ciphered byte.

As it is already mentioned that AES S-box, S, performs bijective mapping, while the generalized logistic map does not perform bijective mapping. It seems very difficult to calculate the difference propagation probability for (5.16), due to its hybrid nature consisting of both bijective and non-bijective maps.

In order to analyze the difference probability for (5.16), we consider both AES S-box S, and the generalized logistic map as a one S-box and refer to it as S' and $DP^{S'}$ is derived from, how the difference propagates through S'. The difference probability for the hybrid S-box S', is calculated as:

$$DP^{S'} = \left(\frac{\#\{x \in X | S'(x) \oplus S'(x \oplus \Delta x) = \Delta y\}}{2^m} \right) \tag{5.17}$$

where $S' = S(x) \oplus f(y_{i-1})$ and $S'(x \oplus \Delta x) = S(x \oplus \Delta x) \oplus f(y_{j-1})$. y_{i-1} and y_{j-1} are the previous outputs corresponding to $S'(x)$ and $S'(x \oplus \Delta x)$ respectively. Initially y_{i-1} and y_{j-1} are considered to be equal to 0.

The distribution of the differentials for the AES S-box and hybrid S-box is shown in Fig. 5.30 and Fig.5.31, respectively. In case of the hybrid S-box, there are only two differentials that occur with the high probability 8/256, while in case of AES S-box, for every input difference Δx, there exists a corresponding output difference Δy, with the maximum probability of 4/256. From Fig. 5.30, it is evident that the distribution of the differentials for AES S-box is box is quite uniform compared to the hybrid S-box. In fact, AES S-box is an optimized S-box. The differential probabilities for few differentials of the hybrid S-box are given in the table 5.6. For the rest of the differentials, the corresponding $DP^{S'} \le 5/256$. During the analysis, it is also observed that the a large number of differentials occur with the probability $DP^{S'} = 1/256$ (ideal case). From the analysis, it is evident that the maximum differential probability of S' is equal to $8/256 = 2^3/256 = 2^{-5}$.

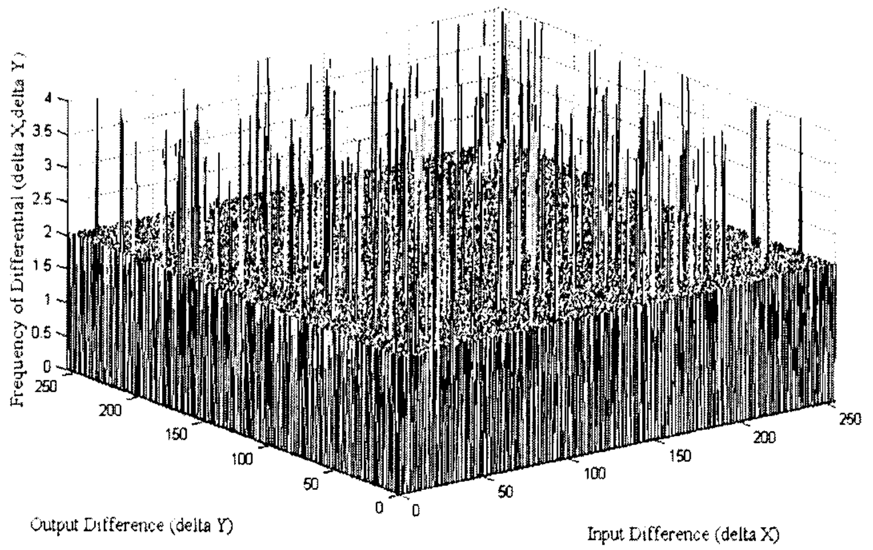


Fig. 5.30. Distribution of differentials AES S-box $(\Delta x, \Delta y)$.

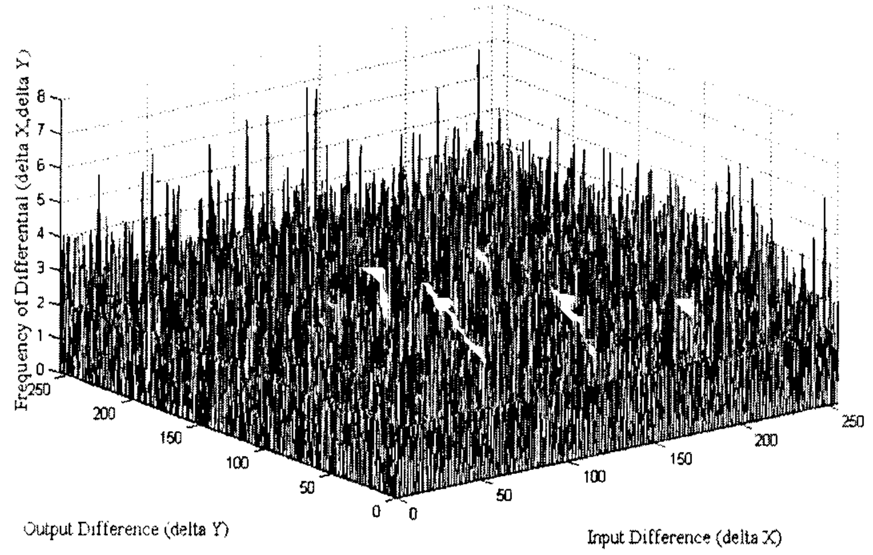


Fig. 5.31. Distribution of differentials for Hybrid S-box $(\Delta x, \Delta y)$.

Table 5.6. Probability of typical differentials in case of hybrid S-box: Differentials with their corresponding frequency of occurrence in case of hybrid S-box

Input Difference	Output Difference	Probability
200	100	8/256
91	162	8/256
219	80	7/256
165	85	7/256
219	92	7/256
240	136	7/256
17	175	7/256
46	181	7/256
30	214	7/256
245	234	7/256
234	6	6/256
152	10	6/256
90	19	6/256
64	20	6/256
209	21	6/256
226	25	6/256
133	26	6/256
72	27	6/256
60	30	6/256
34	35	6/256
175	35	6/256
246	46	6/256
109	49	6/256
64	61	6/256
243	77	6/256
192	83	6/256
10	84	6/256
88	102	6/256
108	102	6/256
99	112	6/256
61	123	6/256
26	128	6/256
155	141	6/256
129	161	6/256
162	166	6/256
158	190	6/256
26	192	6/256
125	196	6/256
71	215	6/256
251	215	6/256
56	227	6/256
50	243	6/256
6	251	6/256

5.3.7.5 Differential Characteristic Probability of H-CES

In this section, the differential characteristic probability for the two rounds of H-CES is calculated. In the upcoming discussion, the number of active S-boxes and effect of each active S-box over all other bytes in the input matrix will be analyzed over two rounds. It is important to highlight two terms "effect of an active S-box" and "number of active S-boxes". "*Effect of an active S-box*" refers to the effect of a current active S-box over the rest of the bytes in the input matrix consisting of 16-bytes, while "number of active S-boxes" refers to the number of active S-boxes, which may or may not include the effect of each bundle or byte in the input matrix. In what follows, the terms 'bundle' and 'byte' will be used interchangeably.

A differential characteristic is a sequence of input and output differences to the rounds so that the output difference from one round corresponds to the input difference for the next. Pairs of differences A' and B' with a large probability $\mathrm{Prob}(A'|B')$ are found by the construction characteristics. An r-round characteristic constitutes an $(r+1)$-tuple of difference patters: $(X_0^{'}, X_1^{'}, \ldots X_r^{'})$. The probability of this characteristic is the probability that an initial difference pattern $X_0^{'}$ propagates to difference patterns $X_1^{'}, X_2^{'}, \ldots X_r^{'}$ after $1, 2, \ldots, r$ rounds, respectively. Under the Markov assumption, propagation probability from $X_{i-1}^{'}$ to $X_i^{'}$ is independent of the propagation from $X_0^{'}$ to $X_{i-1}^{'}$, hence the differential characteristic probability is given as:

$$\prod_i \mathrm{Prob}(X_i^{'} | X_{i-1}^{'}) \qquad (5.18)$$

By computing the minimum number of differentially active S-boxes, the security of a block cipher in terms of practical security against differential cryptanalysis is evaluated. The upper bounds for the maximum differential characteristic probability can be calculated from the minimum number of differentially active S-boxes. Thus, it is important to analyze, how this minimum number of differentially active S-boxes increase by considering diffusion layer in consecutive two rounds (effect of one byte over the rest of bytes).

Lemma 1

For H-CES, there is no round where number of active S-boxes is any less than 16.

Proof of Lemma 1

Visualizing the structure of H-CES, it can be seen that each i^{th} bundle depends on the $(i-1)^{th}$ ciphered bundle, except the first bundle which depends on the last bundle in the input matrix. Hence, at the start of an i^{th} round if there is no active S-box, then, at the end of the i^{th} round, the number of active S-boxes will not be less than 16 (maximum possible), as the effect of each bundle is propagated forward due to the dependence of the each bundle over the previous bundle. Hence the number of active boxes will be 16.

Lemma 2

The only round where the effect of a bundle is diffused over only 1 bundle is the 1^{st} round.

Proof of Lemma 2

Except for the 1^{st} bundle in the input matrix of 16-bundles, it is evident that the i^{th} bundle in the input matrix is dependent on the $(i-1)^{th}$ ciphered bundle.

Hence, each input bundle in the input matrix is dependent on all previous bundles in the input matrix or it can be said that the current state is the result of all previous states in the input matrix. At the same time, 1^{st} bundle in the input matrix is dependent on the last ciphered bundle, which is effected by all other bundles in the input matrix.

Consider now1^{st} round, except for the last bundle, each bundle affects or activates more than one bundle in the input matrix. Since the 1^{st} bundle is encrypted at the end and depends on the last bundle in the input matrix, hence only last bundle affects or activates only 1 bundle in the input matrix in the 1^{st} round. Hence, in the 1^{st} round, the effect of the last bundle in the input matrix is diffused only over one bundle in the 1^{st} round.

Lemma 3

The effect of each active S-box in $(i-1)^{th}$ round (if it exists), is diffused over all bundles (equal to 16) at the end of the i^{th} round (if it exists).

Proof of Lemma 3

From the Lemma 2, each bundle affects more than one bundle, except in case of the last bundle which only affects the 1^{st} bundle in the input matrix at the end of the 1^{st} round. The only bundle in the input matrix that affects maximum number of bundles equal to 15 in the 1^{st} round is the 1^{st} bundle since each bundle is dependent on the previous bundles. Considering the last bundle in the input matrix, which affects only 1^{st} bundle in the 1^{st} round, the effect of last bundle is diffused over rest of the bundles in the input matrix in the next round (2^{nd} round).

Hence the effect of each bundle is diffused over all other bundles in the next i^{th} round (if it exists).

From lemmas 1, 2 and 3, it is proved that the effect of each bundle is diffused over all bundles in the input matrix in two rounds. Hence, the number of active S-boxes is greater than 16 for two rounds. Hence for a two round trail considering the diffusion of each bundle over all other bundles in the input matrix, the maximum differential characteristic probability of H-CES is given by (5.19).

$$DCP_{\max}^{S'} \le \left(2^{-5}\right)^{16} = \left(2^{-80}\right) \tag{5.19}$$

In case of H-CES, the round transformation provides very high diffusion. However, it is worthwhile to remember that transformation that provides high diffusion (equal to number of bundles in the state) have a tendency to have a high implementation cost. In H-CES, the round transformation is not optimized. The affect of each active hybrid S-box is locally propagated forward, as each next bundle is dependent on the previous bundle. However this lowers down the throughput of H-CES, as there is little parallelism – an attribute that is desirable for hardware implementation. The structure of H-CES can be improved further, however, presently, this outside the scope of this dissertation.

From above lemmas, it is seen that within two rounds of H-CES, the affect of each bundle is diffused over rest of the bundles, and the number of active S-boxes within a round are no less than 16. Hence for a four- round differential trail, number of active S-boxes will be 48. According to the four-round propagation theorem – Theorem 9.5.1 in [Daemen & Rijmen, 2002], for AES, the number of active S-boxes in a four-round differential trail or linear trail is lower bounded by 25. Hence for AES, maximum differential characteristic probability for a four-round trail is lower bounded by 2^{-150}, while maximum linear probability for a four-round linear trail is 2^{-75}. Hence for an eight-round trail, the differential and linear probabilities are lower bound by 2^{-300} and

2^{-300}, respectively. For H-CES, since the minimum number of active S-boxes is lower bounded by 48, hence, maximum differential characteristic probability for a four-round differential trail is lower bounded by 2^{-240}.

5.3.2 Related-Key Attacks

Related-key attack is an attack introduced by E. Biham in [Biham, 1994]. Under related-key attacks, the cryptanalyst is assumed to have access to the ciphertexts that result from using keys that are either unknown or partly unknown but are said to have some distinguishing feature (chosen relationship). The cryptanalyst exploits that feature and tries to use a key having similar feature. Usually a cipher is protected from related-key attacks by incorporating a good key scheduling algorithm. The key schedule be derived from a step involving diffusion and non-linearity so as to make this type of attack difficult to mount against it.

In H-CES, secret key stream used for masking is extracted from a stream cipher based on m-PLCMs. In H-CES, an external secret key of 128-bit is mapped to an initial condition (IC) through a sub-algorithm that is based on 16-PLCMS. Each i^{th} 8-bit block from an external secret key of 128-bit is uniformly mapped to a point in the interval $(0,1)$ that serves as IC for corresponding i^{th} PLCM with different control parameter, p_i. Each PLCM is then iterated for a number of times equal to decimal equivalent of the ASCII value of the corresponding 8-bit block. After iterating i^{th} PLCM, the outputs of all PLCMs are summed to get a unique initial condition, IC. Hence different external secret keys result in different ICs.

Under related-key attacks, encryption is carried out with two different keys having a chosen relationship. From the description of key schedule of H-CES, it is seen that two different keys lead to two different initial conditions, ICs. Now, from the property of chaotic dynamical systems, slightly different initial conditions, ICs, result in completely different trajectories. Hence, two different initial conditions, *ICs*, generate totally independent and different secret key streams that are used for masking purposes in H-CES. From the above arguments, it is evident that it is very difficult to mount key-related attacks on H-CES.

5.3.3 Properties of the H-CES

In this section some important attributes of the proposed H-CES are investigated to ensure that H-CES effectively randomizes the input plain data.

1. Average number of 1s in ciphertext bytes if plaintext bytes contains only one bit of 1.
2. Average number of differing bit positions in ciphertext byte when the two plaintext bytes differ in only one bit position.
3. Average number of differing bit positions in ciphertext bytes when the plaintext bytes are all 0s and the two keys differ in only one bit position.

The above three properties basically ensures that H-CES effectively brings in the diffusion in the ciphered data. For evaluation of the above three properties, a random plaintext data of the length 6000 bytes of 8-bit each, is generated using **rand ( )** in MATLAB®.

Fig. 5.32 shows that the plaintext where each byte contains only one bit 1 but after encryption by H-CES, the average number of 1s in a ciphertext byte is almost ideally distributed and the average number of 1s are between 3.9742 & 4.0228, depending on the number of rounds of H-CES. Ideally for an 8 bit byte, this number should be 4.

Fig. 5.33 shows the average number of different bit positions in ciphertext bytes, when the two plaintext bytes differ in only one bit positions. Two list of random plaintexts are generated, where corresponding plaintext bytes in the two lists differs only in the least significant bits i.e. if the least significant bit is ‘1’ in a byte from the first plaintext list, then the least significant bit in the corresponding byte of the second plaintext is ‘0’ and vice versa. The two lists of plaintexts are encrypted by H-CES using the same secret key *K*. The average number of differing bit positions in the corresponding bytes of the two ciphertext lists, ideally 4, is between 3.9680 & 4.0278, depending on the number of rounds of H-CES. This property ensures the sensitivity of H-CES with respect to the plaintext data i.e. if two plaintext bytes differs only by one bit then it will lead to a totally different ciphertexts.

Fig. 5.34 demonstrates the key sensitivity of H-CES. The average number of differing bit positions in two lists of ciphertext bytes are calculated, when the corresponding plaintext bytes are encrypted by two arbitrary secret keys that differs in only one bit. The average number of differing bit positions in the corresponding bytes of two ciphertext lists, ideally 4, is between 3.9667 & 4.0370, depending on the number of rounds of H-CES.

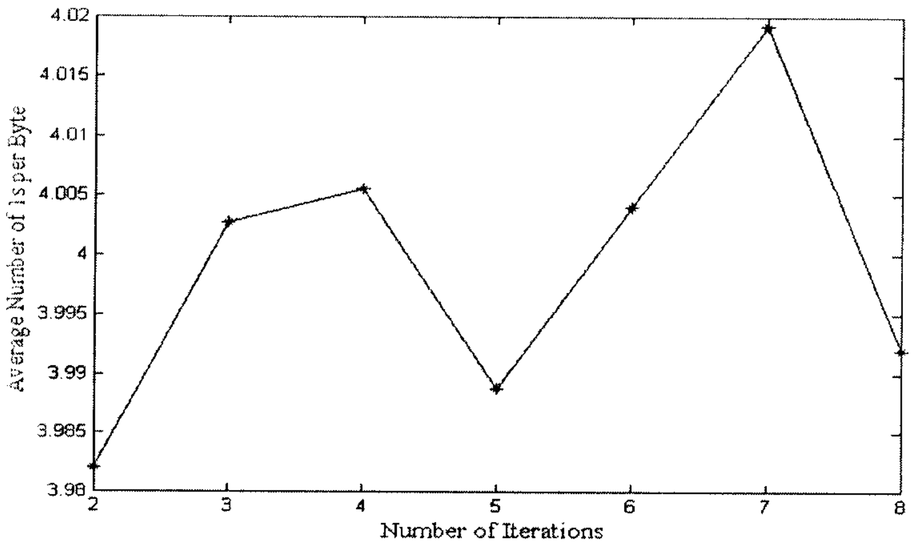


Fig. 5.32. Diffusion property of H-CES: The average number of ones in a ciphertext byte when the plaintext byte contains only one bit for different rounds of H-CES, when encrypted with the secret key {‘a’ ‘b’‘c’‘d’‘e’‘f’‘g’‘h’‘i’‘j’‘k’‘l’‘m’‘n’‘o’‘p’}. In case of AES, the average number of ones in a ciphertext byte are 3.979, when the encryption key is {'00' '01' '02' '03' '04' '05' '06' '07' '08' '09' '0a' '0b' '0c' '0d' '0e' '0f'}.

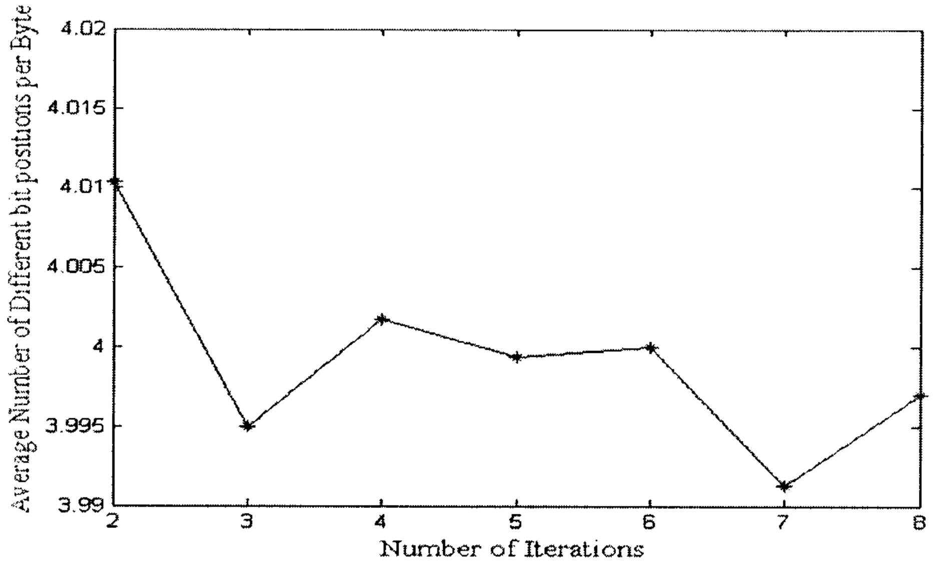


Fig. 5.33. H-CES Sensitivity to plaintext: The average number of different bit positions in two ciphertext bytes when the two plaintext bytes differ in only one bit positions for different rounds of H-CES, where the encryption key used is {'a''b''c''d''e''f''g''h''i''j''k''l''m''n''o''p'}. For AES the figure is 3.9898, when the encryption key is {'00' '01' '02' '03' '04' '05' '06' '07' '08' '09' '0a' '0b' '0c' '0d' '0e' '0f'};

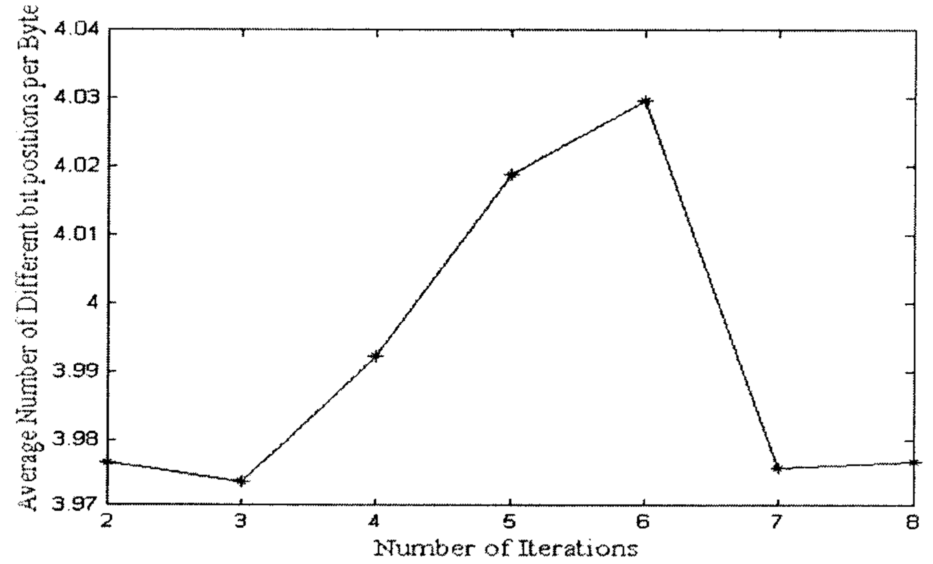


Fig. 5.34. The average number of different bit positions in two ciphertext bytes when the plaintext bytes are all 0s and the two keys differ in only one bit position. Plaintext is first encrypted with secret key {'a' 'b''c''d''e''f''g''h''i''j''k''l''m''n''o''p'}and then with secret key {'a' 'b''c''d''e''f''g''h''i''j''k''l''m''n''o''o'}. For AES the figure is 4.6250, when first encrypted with key {'00' '01' '02' '03' '04' '05' '06' '07' '08' '09' '0a' '0b' '0c' '0d' '0e' '0f'}; and then encrypted with the key {'00' '01' '02' '03' '04' '05' '06' '07' '08' '09' '0a' '0b' '0c' '0d' '0e' '0e'};

5.4 Summary

In this chapter, extensive analysis of both HyChIES and H-CES has been carried out. Analysis of HyChIES is presented first and it is shown that HyChIES effectively randomizes the plain-images and correlation among neighboring pixels of ciphered image is negligible after encryption by HyChIES. HyChIES also fulfills the avalanche property at the level of image and hence, proves sensitive to a single change in one pixel. HyChIES also fulfills key sensitivity requirements and hence one bit of change in secret key leads to totally different ciphered images. It is also shown that HyChIES effectively distribute input differentials (difference between two plain images A and B). In addition, the histogram of a mask image constructed using the ciphered images of A and B is also shown to be uniform. Hence HyChIES is secure against the differential attack to which Pareek et al.'s image cipher is susceptible. Subsequently, we also performed the detailed key space analysis of HyChIES and showed that the size of key space is very large. Mapping of external secret key of 128-bit to an IC does not affect the key space size and the mapping process does not possess the symptoms from which Pareek et al.'s and Fei et al.'s ciphers in [Pareek et al., 2005] [Fei et al., 2005] suffer. We also compare the computational complexity of HyChIES with some recent representative chaotic ciphers [Chen et al., 2004] [Fei et al., 2005] [Pareek et al., 2006] and it is shown that the computational complexity of HyChIES is comparatively very low and, therefore, it is suitable for real time encryption of digital images.

After the analysis of HyChIES, security analysis of H-CES is performed. First the difference propagation probability of each component of H-CES is separately evaluated. Then we obtain the difference propagation probability of hybrid S-box, where hybrid S-box consists of AES S-box, generalized logistic map, and a ciphertext feedback. On the bases of differential probability of the hybrid S-box, differential characteristic probability for two rounds of H-CES is calculated that is equal to $\left(2^{-80}\right)$. It is also shown that it is not possible to mount key-related attacks on H-CES either. At the end, we present some additional properties of H-CES relating to its sensitivity to plaintext and secret key.

In the next chapter, we will draw conclusions for the entire work and suggest future work that can be carried out in continuation of the work reported herein.

6

CHAPTER

Conclusions

6.1 Introduction

In the previous chapters, we presented the detailed survey of the multimedia encryption schemes, followed by the analysis of some recently proposed chaos based encryption schemes. Based on the lessons learnt from the survey and analysis of the recently chaotic schemes, we proposed an efficient hybrid chaotic image encryption schemes, further modified the same as an AES like 128 bit block cipher. In this chapter, we conclude the entire work and suggest future work to further research in this area.

6.2 Conclusions

In this dissertation, we surveyed various multimedia encryption schemes. It is observed that since 1995, chaotic maps have attracted significant attraction for encryption of the digital images due to their extreme sensitivity to initial conditions and control parameters and are considered to be simpler and convenient to implement. Several schemes have been proposed in the literature based on variety of chaotic maps. However, most of the proposed schemes were later reported to be broken. For example CKBA (Chaotic Key

Based Algorithm) proposed in [Cheng & Guo, 2000] is successfully broken in [Li & Zheng, 2002 (b)] with very less computational load due to the fact that the proposed scheme in [Cheng & Guo, 2000] works like a stream cipher, where two keys are dynamically selected for XORing the gray level values of the image. A mask image is constructed from one known plain-image and its corresponding cipher-image, that can then be used for recovering any other plain-images, encrypted with the same secret key K. Cheng et al.'s scheme in [Cheng & Guo, 2000] also lacks the diffusion property, and a change in one pixel does not affect the rest of the pixels in an image. In the Appendix A, of this dissertation, a modified-CKBA is proposed and shown secure against the attack presented in [Li & Zheng, 2002 (b)]. Modified-CKBA includes a ciphertext feedback and a suggestion of minimum number of rounds to be equal to two, instead of one. The ciphertext feedback is very useful for securing a cryptosystem against the known plaintext and chosen plaintext attacks. Due to the ciphertext feedback, M-CKBA fulfills the avalanche property at the level of image.

Pareek et al. recently proposed several encryption schemes, however, all of them have been reported to be insecure. We analyzed Pareek et al.'s recently proposed chaotic block cipher [Pareek et al. 2005]. It is found that proposed chaotic block cipher functions like a stream cipher and is insecure against the known plaintext and chosen-plaintext attacks. Pareek et al. claimed that the key space size of the cryptosystem is 2^{128}, mapped as it is on the external secret key of 128 bits. However, it is shown that the key space size is much less than 2^{128}. Similar to Pareek et al.'s scheme, Fei et al. also proposed an image encryptions scheme in [Fei et al., 2005], based on multiple 3-D chaotic systems, where an external secret key of 128 bit is used and further mapped to an initial condition, IC. It is identified that the key space size is much less than 2^{128} as claimed by Fei et al. because it lacks the positional significance of each character in the secret key. It is shown that the possible number of ICs after the mapping process is much less than 2^{128}.

Pareek et al. recently proposed an image encryption scheme based on chaotic logistic map in [Pareek et al., 2006] and claimed that the proposed scheme fulfills the requirements of the real time secure transmission of digital images. However in the chapter 3 of this dissertation, we analyzed the proposed cipher and pointed out that the proposed scheme is not secure against differential attack and does not fulfill real-time requirement. We also identified a few other security defects in the Pareek et al. image encryption scheme as reported in the thesis. Also their claim of Number of Pixel Change Rate (NPCR) of about 99% is also not correct, as the effect of one pixel is not diffused over rest of the pixels in the image.

Along with security defects, it is also observed that for high security most of the chaotic ciphers rely on relatively high number of iterations, ranging between 0 and 255. For example in Pareek et al.'s ciphers [Pareek et al., 2005] [Pareek et al.'s, 2006], the number of iteration used for an 8-bit block are dependent on the ASCII value of the secret key characters, hence their suitability for the real time encryption of the digital images is questionable. The throughput of the Fei et al. cipher is also very low for the number of iterations is large- between 5 and 29. Fei et al. cipher is also inconvenient to implement as it seeks to use 3-D chaotic maps.

Based on the knowledge so acquired from the surveyed and analyzed chaotic ciphers, we proposed an image encryption scheme which is named as Hybrid Chaotic Image Encryption Scheme (HyChIES). The proposed scheme is based on m-PLCMs, a generalized logistic map, AES S-box and ciphertext feedback. An external secret key of 128 bit is used, further mapped through a sub-algorithm to acquire initial condition for m-PLCMs used for extracting the key stream for masking the gray level values of the image. A generalized logistic map is used for mixing purpose, whose operands are previous ciphertext pixels (ciphertext feedback at the level of pixels) and corresponding 8-bit key stream extracted from m-PLCMs. It is identified that the generalized logistic map behaves like an S-box. However, due to its nature of being two-to one mapping, it is not sufficiently secure against conventional differential cryptanalysis and few of the differentials occur with relatively high probability. In fact, all of the current chaotic maps are many-to-one and, hence, rigorous analysis based on the conventional cryptanalysis techniques may show them susceptible. Hence, in order to map gray level values, it is proposed to use the AES S-box, while mixing and perhaps some stretching is done by the generalized logistic map.

The analysis of the HyChIES shows that it is extremely sensitive to the change in pixels and fulfills the avalanche property at the level of image and effectively randomizes plain images. The number of rounds $n = 4$, is sufficient to achieve higher level of security and maintaining the real time constraints for digital image encryption. The throughput of the proposed scheme for the un-optimized MATLAB® code is calculated to be 2MB/s for the number of rounds equal to 4. Two rounds are sufficient to diffuse the effect of one pixel over the rest of pixels.

Going further, we also proposed an AES like 128-bit block cipher, named as Hybrid-Chaotic Encryption Scheme (H-CES). The heart of both cryptosystems is same. It consists of AES S-box, generalized logistic map and ciphertext feedback. In order to analyze the differential characteristic probability for H-CES, the AES S-box, generalized logistic map and ciphertext feedback are considered as a single hybrid S-box. The propagation of differentials through hybrid S-box is extensively studied. It is found that the maximum differential probability for a differential through the hybrid S-box is 2^{-5}. It is observed that there are only tow differentials $(200, 100)$ and $(91, 162)$, that occurs with this relatively high probability of 2^{-5}. In case of AES S-box, for every differential there exists a corresponding differential that occurs with the high probability of 2^{-6}. In our opinion, it might be good for the hybrid S-box, that only two differentials occur with high probability. It is also observed that a number of differentials occur with the differential probability of 2^{-8} (ideal case). Based on the maximum differential probability of the hybrid S-box, differential characteristic probability for two rounds of H-CES is calculated to be 2^{-80}, as there is no round with number of active S-boxes any less than 16, as each bundle depends on the previous bundle. Hence the proposed H-CES is secure against the differential cryptanalysis.

6.3 Suggested Future Work

- The structure of H-CES can be further improved by bringing in parallelism and multiple ciphertext feedbacks.

- Systematic analysis of discrete chaotic maps is necessary so that mixing and stretching properties are adequately understood. This would enable researchers to design cryptosystems with encryption equivalent to substitution and permutation based conventional ciphers.

- At the same time, new cryptanalysis techniques need to be developed so mixing/stretching based ciphers can be adequately analyzed. In fact this is every important, as we know the recent advances in block cipher cryptology are promoted by the emergence of differential and linear cryptanalysis. This shows the importance of cryptanalysis. This will also help in exploring the relationship between chaotic and conventional cryptography.

- Issues relating to implementation on chips also have to be adequately analyzed. The specifications of hardware and the way it affects performance of encryption algorithms need to be examined in more detail.

As one can see, there are number of research issues relating to the chaos based cryptography that are yet to be answered. The structures of most of the chaotic ciphers are still ad-hoc and their design is based on intuition. There is also a need for investigating the deeper relationship between chaos and cryptography.

This thesis is a step in that direction. In this thesis, it has been shown that significant improvements can be made by trying to marry the conventional cryptography and chaos. A deeper understanding of the relationship and also the properties of chaos relevant to cryptography will definitely prove very helpful.

A

APPENDIX

On Securing CKBA: Ciphertext Feedback based Algorithm

A.1 Introduction

Cheng & Guo proposed the original Chaotic Key-Based Algorithm (CKBA) in 2001. [Cheng & Guo, 2000]. However CKBA has been shown to be insecure against known plaintext attack in [Li & Zheng, 2002 (b)]. Li and Zheng created a mask image from one known plain image and its corresponding cipher image and used this mask image to recover other plain images encrypted with the same secret key K. In what follows, first the original CKBA is described, followed by the detailed description of the Modified CKBA. Later it is proved that the modified-CKBA is secure against the proposed attack in [Li & Zheng, 2002 (b)].

A.2 Chaotic Key Based Algorithm

In this section the original chaotic key based algorithm (CKBA) is presented. Let f denote an image of size $M \times N$ pixels and $f(x,y), 0 \le x \le M-1, 0 \le y \le N-1$, be the grey level of f at position (x,y). The original CKBA consists of the following steps:

Step1

Determine the two keys, $key1$ and $key2$, and the two parameters, M and N, and set $l = 0$.

Step 2

Determine 1-D chaotic system and its initial point $x(0)$. Generate the chaotic sequence $x(0), x(1), x(2), \ldots, x(MN/8-1)$ from the chaotic system and then create $b(0), b(1), \ldots,$ $, b(2MN-1)$ from $x(0), x(1), x(2), \ldots, x(MN/8-1)$ by the generating scheme that $0.b(16n+0)b(16n+1)\ldots b(16n+13)b(16n+14)b(16n+15)$ is the binary representation of $x(n)$ for $n = 0, 1, 2, \ldots, (MN/8-1)$.

Step 3

```
For $x = 0$ to $M - 1$
    For $y = 0$ to $N - 1$
    Switch $(2 \times b(l) + b(l))$
        Case 3:    $f'(x,y) = f(x,y) \oplus$ XOR $key1$
        Case 2:    $f'(x,y) = f(x,y) \oplus$ XNOR $key1$
        Case 1:    $f'(x,y) = f(x,y) \oplus$ XOR $key2$
        Case 0:    $f'(x,y) = f(x,y) \oplus$ XNOR $key2$
    $l = l + 2;$
    End
End
```

Step 4

After result f' is obtained, stop the algorithm.

$key1$ and $key2$, are 8-bit values, used dynamically for XORing or XNORing the grey level values of the image. Let $key1 = \sum_{i=0}^{7} a_i \times 2^i$ and $key2 = \sum_{i=0}^{7} d_i \times 2^i$. The basic criterion to select $key1$ and $key2$, is given in (A.1).

$$\sum_{i=0}^{7} (a_i \oplus d_i) = 4. \tag{A.1}$$

Which implies only those values of a_i and d_i are allowed whose sum after XORing corresponding values of a_i and d_i, is equal to 4.

A.2 Modified- Chaotic Key Based Algorithm

In this section, a modified version of CKBA is presented, called M-CKBA. The M-CKBA is shown to be secure against the proposed attack of [Li & Zheng, 2002 (b)]. In M-CKBA, cipher text feed backs are introduced and the number of rounds n are suggested to be 2. Also an efficient random number generated is suggested, instead on random number generator based on single 1-D chaotic system. In what follows, we present the M-CKBA briefly, and follow that up with the results of its analysis:

A.3.1 M-CKBA Encryption Algorithm

Stepl

This is the initialization step of M-CKBA. First Determine the two keys, $key1$ and $key2$ for each round, and the two parameters, M and N, and set $l = 0$. Next, similar to the original CKBA, $key1$ and $key2$, are used to XOR or XNOR the grey level of each pixel randomly according to a pseudo-random bit generator. However the pseudo-random bit generator is based on multiple 1-D chaotic systems instead of a single chaotic system. The two keys are selected according to the criterion given in (A.1).

After determining $key1$ and $key2$, initial condition IC is determined. This IC serves as master initial condition and is used to determine 2nd initial condition called as IC_2. The master initial condition will be referred to as IC_1. Let IC_1 be represented, in decimal equivalent, by $0.k_1k_2\cdots k_n$, then IC_2 is determined in decimal equivalent as:

$$IC_2 = 0.(9-k_1)(9-k_2)\cdots(9-k_n). \tag{A.2}$$

where k_i represents the corresponding digit in IC_1.

The encryption of the image is performed in the step 3. After encrypting the grey level of the current pixel, it is stored in the variable C, which is used in the encryption of the next grey level of the pixel. Initially the value of C is equal to 0. The ciphertext feedback diffuses the effect of the each pixel over all other pixels and provides the said avalanche property at the level of image. Also due to the said feedback, M-CKBA is secure against the proposed attack in [Li & Zheng, 2002 (b)].

Step 2

In the original CKBA, *key*1 and *key*2, are dynamically selected for XORing or XNORing the grey level of each pixel according to the Pseudo- Random bit generator based on a single 1-D chaotic map. However the cryptosystem based on single chaotic systems are considered to be insecure. In M-CKBA, a pseudo random bit generator based on triple PLCMs is used. The employed pseudo random bit generator in M-CKBA is called as Chaotic Pseudo Random Bit Generator C-PRBG, which is also used in [Ntalianis and Kollias, 2005]. In C-PRBG, the three PLCMs are denoted by $F_1(x_1, p_1), F_1(x_2, p_2)$, and $F_3(x_3, p_3)$:

$x_1(i+1) = F_1(x_1(i), p_1), x_2(i+1) = F_2(x_2(i), p_2)$, and $x_3(i+1) = F_3(x_3(i), p_3)$, where p_1, p_2, p_3 are three different control parameters, $x_1(0), x_2(0)$ and $x_3(0)$ are initial conditions and $\{x_1(i)\}, \{x_2(i)\}, \{x_3(i)\}$ denote the three chaotic orbits. The pseudo-random bit sequence can be defined as:

$$b(i) = \begin{cases} 1, & F_3(x_1(i), p_3) > F_3(x_2(i), p_3) \\ b(i-1) & F_3(x_1(i), p_3) = F_3(x_2(i), p_3) \\ 0, & F_3(x_1(i), p_3) < F_3(x_2(i), p_3) \end{cases} \tag{A.3}$$

where $b(i)$ represents the generated pseudo-random bit sequence.

According to this scheme the generation of each bit is controlled by the orbit of the third chaotic system, having as initial conditions the outputs of the other two chaotic systems. In the first round of the encryption, the initial conditions $x_1(0)$ and $x_2(0)$ is equal to IC_1, while in the 2nd round it is equal to IC_2.

Step 3

In step 3, the encryption of the image is performed. The encryption process of M-CKBA is shown below.

```
For $n = 0$ to 1
    For $x = 0$ to $M - 1$
        For $y = 0$ to $N - 1$

        Switch $(2 \times b(1) + b(2))$

            Case 3:    $f'(x, y) = f(x, y)$ XOR (C) XOR $key1$
            Case 2:    $f'(x, y) = f(x, y)$ XOR (C) XNOR $key1$
            Case 1:    $f'(x, y) = f(x, y)$ XOR (C) XOR $key2$
            Case 0:    $f'(x, y) = f(x, y)$ XOR (C) XNOR $key2$

        $C = f'(x, y)$.
        End
    End
End
```

Step 4

The result f' is obtained and the algorithm is stopped.

A.3.2 Analysis of M-CKBA

In this section, it will be shown that the M-CKBA is secure against the proposed attack in [Li & Zheng, 2002 (b)] and the M-CKBA is more secure than original CKBA [Cheng & Guo, 2000]. The proposed system also fulfills the avalanche property at the level of image. However the computational complexity of M-CKBA is slightly higher compared to CKBA [Cheng & Guo, 2000].

In [Li & Zheng, 2002 (b)], know-plaintext attack is proposed based on the very simple structure of CKBA [Cheng & Guo, 2000]. In the proposed attack one plain-image and its cipher-image are needed to construct a mask image f_m. The mask image is constructed as:

$$f_m(x,y) = f(x,y) \oplus f'(x,y). \tag{A.5}$$

where f and f' are the corresponding plain and cipher images respectively.

Since $a \,\text{XOR}\, b = a \,\text{XNOR}\, \bar{b}$, hence f_m can be used to decrypt any other cipher image encrypted with the same secret key K, due to the fact that it consists of the all the four possible key values used in the encryption of the image i.e. $key1, \overline{key1}, key2, \text{and} \overline{key2}$. However the proposed M-CKBA is secure against such a simple attack. In what follows, it is proved that the M-CKBA is secure against proposed attack due to multiple numbers of rounds and ciphertext feedback (dependence of the current pixel on the previous ciphered pixel) used in the proposed M-CKBA. Also an efficient chaotic pseudo-random bit generator is suggested instead of the pseudo-random bit generator based on single 1-D chaotic system used in CKBA.

Figs. A.1, A.2 shows the plain-image and its corresponding cipher-image respectively that are used to construct a mask image f_m, shown in the Fig. A.3. Fig. A.4 shows another available cipher image encrypted with the same secret key K. The decrypted image using the mask image is shown in Fig. A.5, while the corresponding histogram is shown in the Fig. A.6. It is evident that it is not easy to decrypt the ciphered image in Fig.

A.4 using the mask image in f_m, and the histogram of the decrypted image is uniform, provides no clue to employ any statistical attack on the proposed M-CKBA.

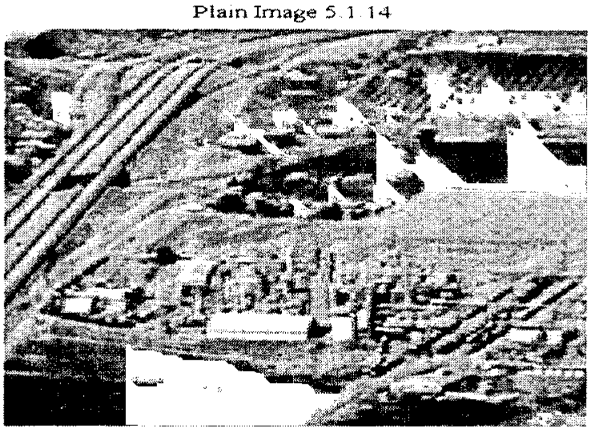


Fig. A.1. f :Plain image 5.1.14.tiff

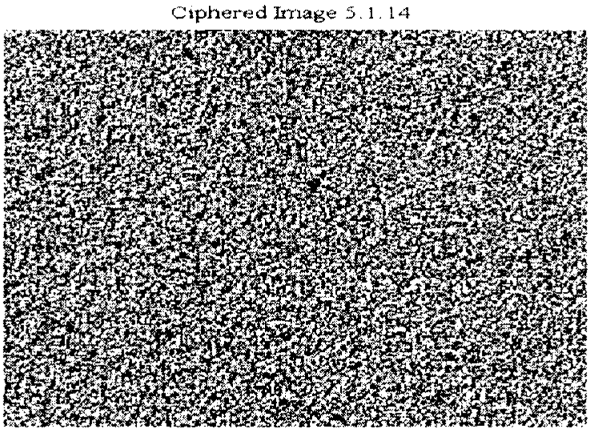


Fig. A.2. f':Corresponding cipher image of 5.1.14.tiff: It is encrypted with the secret key K

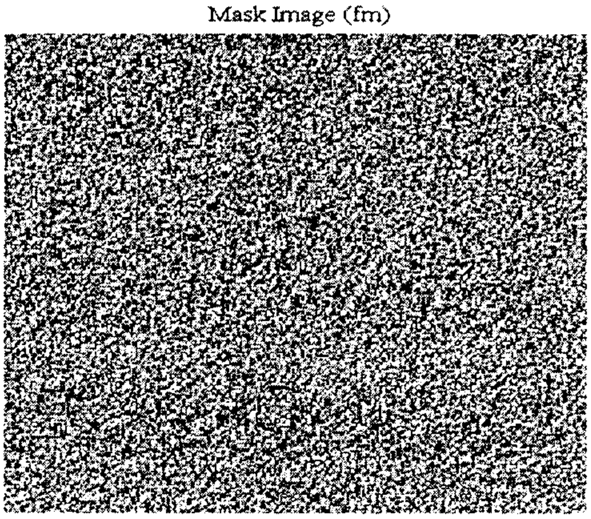


Fig. A.3. f_m :Mask image obtained from f and f'

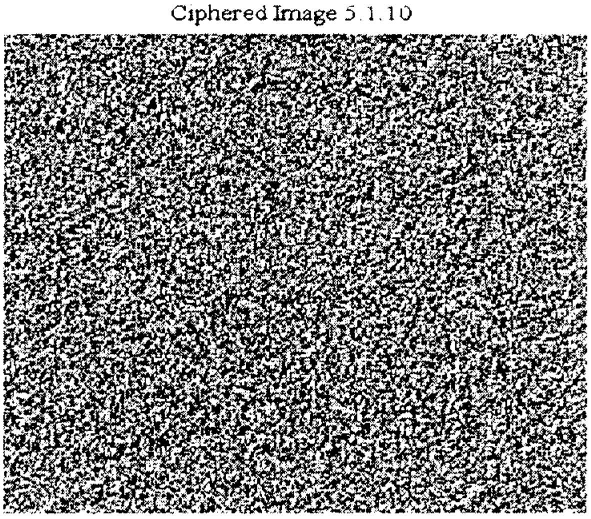


Fig. A.4. f':Corresponding cipher image of 5.1.10.tiff: It is encrypted with the secret key K

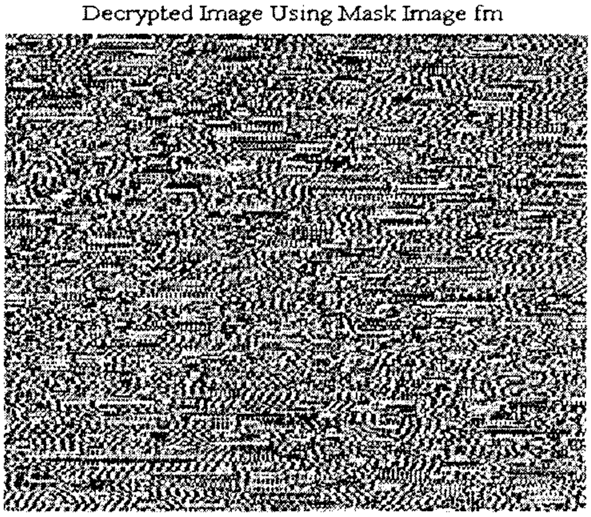


Fig. A.5. f_d :Decrypted image using the mask image f_m

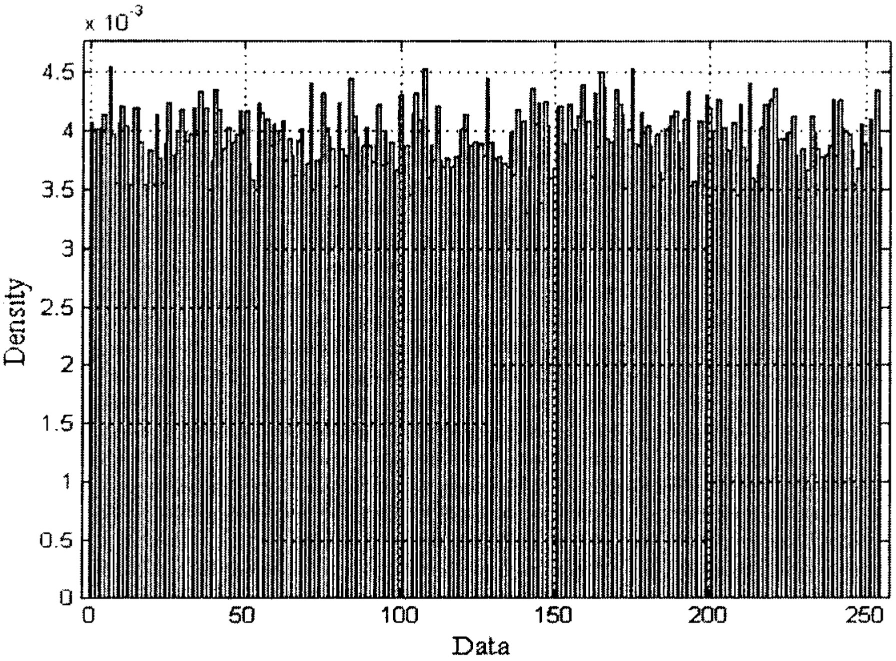


Fig. A.6. Histogram of the decrypted image f_d, using the mask image f_m

A.4 Summary

In this appendix, we showed that original CKBA proposed in [Cheng & Guo, 2000] can be made secure by the inclusion of the ciphertext feedback at the pixel level and using multiple rounds of encryption. Instead of logistic map, we suggested using pseudo-random number generator based on triple chaotic piecewise linear chaotic maps. The throughput of Modified-CKBA compared to original CKBA will be lower due to multiple rounds of encryption and ciphertext feedback.

REFERENCES

[Alattar & Al-Regib, 1999] Alattar, A.M. & Al-Regib, G.I. (1999). Evaluation of Selective Encryption Techniques for Secure Transmission of MPEG-Compressed Bit-Streams. Proceedings of IEEE International Symposium on Circuits and Systems, pp. 340-343.

[Balasubramanian, 2005] Balasubramanian, S. (2005). Image Encryption Using Infinite Series Convergence. 18th International Conference on Systems Engineering, pp. 257 – 262.

[Belkhouche & Qidwai, 2003] Belkhouche, F. & Qidwai, U. (2003). Binary Image Encoding Using 1-D Chaotic Maps, IEEE Region 5 Annual Technical Conference, pp. 39 – 43.

[Beritelli et al., 2000] Beritelli, F., Cola, E.D., Fortuna, L. & Italia, F. (2000). Multilayer Chaotic Encryption for Secure Communication in Packet Switching Networks. IEEE, International Conference on Communication Technology, Vol 2, pp.1575 - 1582.

[Biham, 1991] Biham, E. (1991). Cryptanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT'91. In Proceedings of Advances in Cryptology—EUROCRYPT' 91. Berlin, Germany: Springer-Verlag, pp. 532–534.

[Biham, 1994] Biham, E. (1994). New Types of Cryptanalytic Attacks Using Related Keys. Advances in Cryptology, Proceedings of Eurocrypt'93, LNCS 765, T. Helleseth, Ed., Springer-Verlag, pp.398-409.

[Biham & Shamir, 1991] Biham, E. & Shamir, A. (1991). Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology, Vol 4, no. 1, pp. 3-72.

[Bose & Pathak, 2006] Bose, R. & Pathak, S. (2006). A Novel Compression and Encryption Scheme Using Variable Model Arithmetic Coding and Coupled Chaotic System, IEEE Transactions on Circuits and Systems-I: Regular Papers, Vol 53, Issue 4, pp. 848 – 857.

[Cheng & Li, 2000] Cheng, H. & Li, X. (2000). Partial Encryption of Compressed Images and Video. IEEE Transactions on Signal Processing, Vol 48(8), pp. 2439 - 2451.

[Chen et al., 2004] Chen, G., Mao, Y.B. & Chui, C.K. (2004). A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps. International Journal of Chaos, Solutions and Fractals, pp. 749 - 761.

[Cheng & Guo, 2000] Cheng, J. & Guo, J.I. (2000). A New Chaotic Key-Based Design for Image Encryption and Decryption. IEEE International Symposium on Circuits and Systems, Vol 4, pp. 49 – 52.

[Cheong et al., 2005] Cheong, I-K., Huang, Y-C., Tung, Yi-S., Ke, S-R. & Chen, W-C. (2005). An Efficient Encryption Scheme for MPEG Video. International Conference on Consumer Electronics, pp. 61 – 62.

[Chiaraluce et al., 2002] Chiaraluce, F., Ciccarelli, L., Gambi, E., Pierleoni, P. & Reginelli, M. (2002). A New Chaotic Algorithm for Video Encryption. IEEE Transactions on Consumer Electronics, 48(4), pp.838 – 844.

[Daemen & Rijmen, 2002] Daemen, J. & Rijmen, V. (2002). The Design of Rijndael–AES–The Advanced Encryption Standard, Springer-Verlag.

[Dang & Chau, 2000] Dang, P.P & Chau, P.M. (2000). Image Encryption for Secure Internet Multimedia Applications. IEEE Transactions on Consumer Electronics, Vol 46, pp. 395 – 403.

[Droogenbroeck & Benedett, 2002] Droogenbroeck, M., & Benedett, R. (2002). Techniques for a selective encryption of uncompressed and compressed images. Advanced Concepts for Intelligent Vision Systems, pp. 90 - 97.

[Fei et al., 2005] Fei, P., Qui, S.S. & Min, L., (2005). An Image Encryption Algorithm Based on Mixed Chaotic Dynamic Systems and External Key. IEEE, Proceedings of International Conference on Communications, Circuits and Systems, Vol 2, pp.1135 - 1139.

[Feng et al., 2006] Feng, Y., Li, L. & Huang, F. (2006). A Symmetric Image Encryption Approach Based on Line Maps. IEEE, 1st International Symposium on Systems and Control in Aerospace and Astronautics, pp. 1362-1367.

[Fridrich, 1997] Fridrich, J. (1997). Image encryption based on chaotic maps. IEEE, International Conference on Systems, Man, and Cybernetics, Vol 2, pp. 1105 – 1110.

[Fridrich, 1998] Fridrich, J. (1998). Symmetric ciphers based on two dimensional chaotic maps. International Journal of Bifurcation and Chaos, Vol 8(6): pp. 1259-1284.

[Gluckstad et al., 1999] Gluckstad, J., Mongense, P.C., Toyoda, H. & Hara, T. (1999). Binary Phase Image Encryption Method, IEEE, The Pacific Rim Conference on Lasers and Electro-Optics, Vol 4, pp. 1314 – 1315.

[Guo & Yen, 1999] Guo, J-I. & Yen, J-C. (1999). A New Mirror-like Image Encryption Algorithm and its VLSI Architecture. Proceedings of 10th VLSI Design/CAD Symposium, pp. 327 – 330.

[Hou & Wang, 2003] Hou, Q. & Wang, Y. (2003). Security Traffic Image Transmission Based on EZW and AES, IEEE, Proceedings of Intelligent Transportation Systems, Vol 1, pp 86 – 89.

[Habutsu et al., 1991] Habutsu, T., Nishio, Y., Sasase, I. and Mori, S. (1991). A Secret Key Cryptosystem by Iterating a Chaotic Map. In Proceedings of Advances in Cryptology—EUROCRYPT' 91. Berlin, Germany: Springer-Verlag, pp. 127–140.

[Kankanhalli & Guan, 2002] Kankanhalli, M. S. & Guan, T.T. (2002). Compressed-domain Scrambler/Descrambler for Digital Video. IEEE Transactions on Consumer Electronics, 48(2), pp. 356 – 365.

[Kunkelmann, & Reinema, 1997] Kunkelmann, T. & Reinema, R. (1997). A Scalable Security Architecture for Multimedia Communication Standards, IEEE International Conference on Multimedia Computing and Systems, pp. 660-661.

[Kocarev et al., 1998] Kocarev, L., Jakimosk, G., Stojanovski, T. & Parlitz, U. (1998). From Chaotic Maps to Encryption Schemes. IEEE International Symposium on Circuits and Systems, Vol 4, pp 514 – 517.

[Li et al., 2001] Li. S., Mou, X. & Cai, Y. (2001). Pseudo-Random Bit Generator Based on Couple Chaotic Systems and its Applications in Stream Cipher Cryptography, INDOCRYPT '01 and Lecture Notes in Computer Science, pp. 316 – 329.

[Li et al., 2004] Li, S., Chen, G. & Zheng, X., (2004). Chaos-Based Encryption for Digital Images and Videos. pp. 133-167, ISBN 0849327733, © CRC Press LLC.

[Li & Zheng, 20029 (a)] Li, S. & Zheng, X. (2002) (a). On the Security of an Image Encryption Method. Proceedings of IEEE International Conference on Image Processing, Vol 2, pp. 925 – 928.

[Li & Zheng, 2002 (b)] Li, S. & Zheng, X. (2002) (b). Cryptanalysis of a Chaotic Image Encryption Method. IEEE International Symposium on Circuits and Systems, Vol 2, pp. II-708 - II-711.

[Lian & Wang, 2003] Lian, S. & Wang, Z. (2003). Comparison of Several Wavelet Coefficient Confusion Methods Applied in Multimedia Encryption. International Conference on Computer Networks and Mobile Computing, pp. 372 – 376.

[Lian et al., 2004 (a)] Lian, S., Sun, J. & Wang, Z. (2004) (a), A Novel Image Encryption Scheme Based-on JPEG Encoding, IEEE, Proceedings of 8th International Conference on Information Visualization, pp. 217 – 220.

[Lian et al., 2004 (b)] Lian, S., Sun, J., Wang, Z. & Dai, Y. (2004) (b), A Fast Video Encryption Scheme Based on Chaos, IEEE, 8th International Conference on Control, Automation, Robotics and Vision, Vol 1, pp. 126 – 131.

[Lian et al., 2004 (c)] Lian, S., Wang, Z. & Sun, J. (2004) (c). A Fast Video Encryption Scheme Suitable for Network Applications. IEEE, International Conference on Communications, Circuits and Systems, Vol 1, pp. 566 – 570.

[Lian et al., 2004 (d)] Lian, S., Sun, J. & Wang, Z. (2004) (d). Perceptual Cryptography on JPEG2000 Compressed Images or Videos. The 4th International Conference on Computer and Information Technology, pp. 78 – 83.

[Liu & Li, 2004] Liu, Z., & Li, X. (2004), Motion Vector Encryption in Multimedia Streaming, IEEE, Proceedings of the 10th International Multimedia Modeling Conference, pp. 64 – 71.

[Martin et al., 2005] Martin, K., Lukac, K.R. & Plataniotis, K.N. (2005). Efficient Encryption of Compressed Color Images, Proceedings of the IEEE International Symposium on Industrial Electronics, Vol 3, pp. 1245 – 1250.

[Naor, 1994] Naor, A. (1994). Visual Cryptography. Advances in Cryptology: Eurocrypt '94, Lecture Notes in Computer Science, Vol 950, pp. 1-12.

[Ntalianis & Kollias, 2005] Ntalianis, K.S., & Kollias, S.D. (2005). Chaotic Video Objects Encryption based on Mixed Feedback Multiresolution Decomposition and Time-Variant S-boxes, IEEE International Conference on Image Processing, Vol 2, Issue 11-14, pp. II - 1110-13.

[Pareek et al., 2005] Pareek, N.K., Patidar, V & Sud, K.K. (2005). Cryptography using Multiple One Dimensional Chaotic Maps. Communications in Nonlinear Science and Numerical Simulation, Vol 10, Issue 7, pp: 715 - 723.

[Pareek et al., 2006] Pareek, N.K., Patidar, V & Sud, K.K. (2006). Image Encryption Using Chaotic Logistic Map. Image and Vision computing, http: //www.sciencedirect.com.

[Peitgen et al., 2004] Peitgen, Jürgens & Saupe. (2004). Chaos and Fractals-New Frontiers of Science, Second Editon, Springer-Verlag.

[Pichler & Scharinger, 1996] Pichler, F. & Scharinger, J. (1996). Finite Dimensional Generalized Baker Dynamical Systems for Cryptographic Applications. In Proceedings of 5th International Workshop on Computer Aided Systems Theory (EuroCAST'95), Vol 1030 of Lecture Notes in Computer Science, pp. 465–476.

[Phan & Siddiqi, 2006] Phan, R.C.-W & Siddiqi, M.U. (2006). A Framework for Describing Block Cipher Cryptanalysis. IEEE Transactions on Computers, Vol 55, Issue 11, pp. 1402 – 1409.

[Qiao & Nahrstedt, 1997] Qiao, L. & Nahrstedt, K. (1997). A New Algorithm for MPEG Video Encryption. In Proceedings of the First International Conference on Imaging Science, Systems and Technology, pp. 21 - 29.

[Qiao & Tam, 1997] Qiao, L. & Tam, M-C. (1997). Is MPEG Encryption by Using Random List Instead of Zigzag Order Secure?. IEEE International Symposium on Consumer Electronics, pp. 226 – 229.

[Qi et al., 2000] Qi, D., Zou, J. & Han, X. (2000). A New Class of Scrambling Transformation and its Application in the Image Information Covering. Science in China - Series E (English Ed.), 43(3), pp. 304–312.

[Said, 2005] Said, A. (2005). Measuring the Strength of Partial Encryption Schemes. IEEE International Conference on Image Processing, Vol 2, pp. II - 1126-9.

[Salleh et al., 2002] Salleh, M., Ibrahim, S. & Isinn, I.F. (2002) (a). Ciphering Key of Chaos Image Encryption. Proceedings of International Conference on Artificial Intelligence in Engineering and Technology, pp. 58–62.

[Salleh et al., 2003] Salleh, M., Ibrahim, S. & Isinn, I.F (2003). Enhanced Chaotic Image Encryption Algorithm based on Baker's map. In Proceedings of the International Symposium on Circuits and Systems, Vol 2, pp. 508 - 511.

[Scharinger, 1997] Scharinger. J. (1997). Fast Encryption of Image Data using Chaotic Kolmogorov Flows. In Storage and Retrieval for Image and Video Databases, Vol 3022 of Proceedings of SPIE, pp. 278–289.

[Scharinger, 1998] Scharinger, J. (1998). Secure and Fast Encryption using Chaotic Kolmogorov Flows. In Proceedings IEEE Information Theory Workshop, pp. 124–125.

[Seo et al., 2003] Seo, Y-H., Kim, D-W. Yoo, J-S., Dey, S. & Agrawal, A., (2003). Wavelet Domain Image Encryption by Subband Selection and Data Bit Selection. Proceedings of World Wireless Congress.

[Servetti & Martin, 2002 (a)] Servetti, A. & Martin, J. (2002) (a). Perception-Based Partial Encryption of Compressed Speech. IEEE Transactions on Speech Audio Processing, Vol 10, Issue 8, pp. 637 – 643.

[Servetti & Martin, 2002 (b)] Servetti, A., & Martin, J. (2002) (b). Perception-Based Selective Encryption of G.729 Speech. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol 1, pp. I-621 - I-624.

[Servetti et al., 2003] Servetti, A., Testa, C. & Martin, J.C.D. (2003). Frequency-Selective Partial Encryption of Compressed Audio. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol 5, pp. V - 668-71.

[Shannon, 1949] Shannon, C.E. (1949). Communication Theory of Secrecy Systems, Bell System Technology. Journal.

[Shi & Bhargava, 1998 (a)] Shi, C., & Bhargava, B. (1998) (a). A Fast MPEG Video Encryption Algorithm. Proceedings of 6th ACM International Conference on Multimedia.

[Shi & Bhargava, 1998 (b)] Shi, C., & Bhargava, B. (1998) (b). An Efficient MPEG Video Encryption Algorithm. Proceedings of 17th IEEE Symposium on Reliable Distributed Systems, pp. 381 – 386.

[Shi et al., 1999] Shi, C., Wang, S. & Bhargava, B. (1999). MPEG Video Encryption in Real-Time Using Secret Key Cryptography. Proceedings of PDPTA'99, Vol 6.

[Shieh, 2003] Shieh, J.-R. J. (2003). On the Security of Multimedia Video Information. Proceedings of IEEE 37th Annual International Conference on Carnahan Security Technology, pp. 51 – 56.

[Simmons, 1992] Simmons, G.J. (1992). Contemporary Cryptography. IEEE Press, Piscataway.

[Spanon & Maples, 1996] Spanos, G.A. & Maples, T.B. (1996). Security for Real-Time MPEG-I Compressed Video in Distributed Multimedia Applications. Proceedings of the IEEE 15th Annual International Conference on Computers and Communication, pp. 72-78.

[Sobhy & Shehata, 2001 (a)] Sobhy, M.I. & Shehata, A.-E.R. (2001) (a). Methods of Attacking Chaotic Encryption and Countermeasures. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol 2, pp. 1001 – 1004.

[Sobhy & Shehata, 2001 (b)] Sobhy, M.I. & Shehata, A.-E.R. (2001) (b). Chaotic Algorithms for Data Encryption. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol 2, pp. 997 – 1000.

[Sudharsanan, 2005] Sudharsanan, S. (2005). Shared Key Encryption of JPEG Color Images. IEEE Transactions on Consumer Electronics, Vol 51, pp.1204 - 1211.

[Stallings, 1998] Stallings, W. (1998). Cryptography and Network Security- Principles and Practices., Prentice Hall.

[Tang, 1996] Tang, T. (1996). Methods for Encrypting and Decrypting MPEG Video Data Efficiently. Proceedings of the Fourth ACM International Multimedia Conference, pp. 219 - 230.

[Thorwirth et al., 2000] Thorwirth, N., Horvatic, P., Weis, R. & Zhao, J. (2000). Security Methods for MP3 Music Delivery. Record of the 34th Asilomar Conference on Signals, Systems, and Computers, Vol 2, pp. 1831 – 1835.

[Tosun & Feng, 2000] Tosun, A. S., & Feng, W-C. (2000). Efficient Multi-Layer Coding and Encryption of MPEG Video Streams. IEEE International Conference on Multimedia and Expo, Vol 1, pp. 119 – 122.

[Tosun & Feng, 2001] Tosun, A. S., & Feng, W-C. (2001). Lightweight Security Mechanisms for Wireless Video Transmission. Proceedings of International Conference on Information Technology: Coding and Computing, pp. 157 – 161.

[Uehara, 2001] Uehara, T. (2001). Combined Encryption and Source Coding. http://www.uow.edu.au/~tu01/ CESC.html.

[Vancea, et al., 2005] Vancea, F., Vancea, C. & Borda, M. (2005). Improving an Encryption Scheme Adapted to JPEG Stream Encoding, 7th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, Vol 1, pp. 201-204.

[Wen et al., 2002] Wen, J., Severa, M., Zeng, W., Luttrell, M.H. & Jin, W. (2002). A format-compliant configurable encryption framework for access control of video. IEEE Transactions on Circuits and Systems for Video Technology, 12(6), pp. 545–557

[Wu & Kuo, 2001] Wu, C.-P. & Kuo, C.-C.J. (2001). Fast encryption methods for audiovisual data confidentiality. In Multimedia Systems and Applications III, Vol 4209 of Proceedings of SPIE, pp. 284–295.

[Wu & Kuo, 2005] Wu, C-P., & Kuo, C-C.J. (2005). Design of Integrated Multimedia Compression and Encryption Systems. IEEE Transactions on Multimedia, Vol 7, Issue 5, pp. 828 – 839.

[Wu & Kuo, 2002] Wu, M. & Mao, Y. (2002). Communication - Friendly Encryption of Multimedia. IEEE, Work shop on Multimedia Signal Processing, pp. 292 - 295.

[Yen & Guo, 1999] Yen, J-C. & Guo, J-I. (1999). A New Image Encryption Algorithm and its VLSI Architecture. Proceedings of IEEE Workshop on Signal Processing Systems, pp. 430 – 437.

[Yen & Guo, 2002] Yen, J-C., Guo, J-I. (2002). Design of a New Signal Security System. Proceedings of IEEE International Symposium on Circuits and Systems, Vol 4, pp. 121 – 124.

[Yi et al., 2001] Yi, X., Tan, C.H., Slew, C.K. & Syed, M.R (2001). Fast Encryption for Multimedia, IEEE Transactions on Consumer Electronics, Vol 47, Issue 1, pp. 101 – 107.

[Youssef & Tavares, 2003] Youssef, A.M. & Tavares, S.E. (2003). Comments on the Security of Fast Encryption Algorithm for Multimedia (FEA-M), IEEE Transactions on Consumer Electronics, Vol 49, Issue 1, pp. 168 – 170.

[Zhen & Lei, 2003] Zeng, W. & Lei, S. (2003). Efficient Frequency Domain Selective Scrambling of Digital Video. IEEE Transactions on Multimedia, Vol 5, Issue 1, pp. 118 – 129.

[Zhang et al., 2004] Zhang, M–R., Shao, G-C. & and Yi, K-C. (2004). T-matrix and its Applications in Image Processing. Electronics Letters, Vol 40, Issue 25, pp. 1583 – 1584.

PUBLICATIONS

1. Asim, M. & Jeoti, V. (2007). On Image Encryption: Comparison between AES and a Novel Chaotic Encryption Scheme. IEEE International Conference on Signal Processing Communication and Networking, pp.65-69.

2. Asim, M. & Jeoti, V. (2007). An Efficient Hybrid Chaotic Image Encryption Scheme. The International Conference on New Technologies, Mobility and Security, Paris, France, May 2-4, 2007.

3. Asim, M. & Jeoti, V. (2007). On the Security of a Recent Chaotic Cipher. IEEE, 3rd International Colloquium on Signal Processing and its Applications, pp.293-296.

4. Asim, M. & Jeoti, V. (2007). Comments on Fei-Qui-Min Chaotic Cipher. IEEE, 3rd International Colloquium on Signal Processing and its Applications, pp.297-298.

5. Li, C., Li, S., Asim, M., (2007). On the Security of an Image Encryption Scheme. To appear in Journal of Image and Vision Computing.

6. Asim, M. & Jeoti, V. (2007). Hybrid Chaotic Image Encryption Scheme (HCIES) based on S-box and ciphertext feedback. To appear in IEEE International Conference on Intelligent and Advanced Systems, pp.xx-xx.

7. Asim, M. & Jeoti, V. (2007).On Improving an image encryption schemes based on chaotic logistic map. To appear in IEEE International Conference on Intelligent and Advanced Systems, pp. xx-xx.