Asymmetric Image Encryption based on Cipher Matrices

Sukant Kumar Chhotaray
Roll- 507EC004

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Asymmetric Image Encryption based on Cipher Matrices

Dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

SUKANT KUMAR CHHOTARAY

(Roll- 507EC004)

under the guidance of

Prof. G.S. RATH

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India

February 2012

Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Dr. G. S. Rath
Professor

July , 2014

Certificate

This is to certify that the thesis entitled **Asymmetric Image Encryption based on Cipher Matrices** by Sukant Kumar Chhotaray, submitted to the National Institute of Technology,Rourkela for the degree of Doctor of Philosophy, is a record of an original research work carried out by him in the department of Electronics and Communication Engineering under my supervision. I believe that the thesis fulfills part of the requirements for the award of degree of Doctor of Philosophy. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

G. S. Rath

Acknowledgement

This dissertation, though an individual work, has benefited in various ways from several people. Whilst it would be simple to name them all, it would not be easy to thank them enough.

The enthusiastic guidance and support of *Prof. Girija Sankar Rath* inspired me to stretch beyond my limits. His profound insight has guided my thinking to improve the final product. My solemnest gratefulness to him.My sincere thanks to my co-guide Prof. M.P.Teredesai without whose support my work would have been half done.

I am also grateful to *Prof. S. K. Patra* and *Prof. S. K. Behera* for their ceaseless support throughout my research work. My sincere thanks to *Prof. S. Meher* for his advice. I am grateful to Prof. S. K. Sarangi, Director NIT Rourkela for his kind support.

It is indeed a privilege to be associated with people like *Prof. A. K. Sahoo*,*Nihar Ranjan Biswal*. They have made available their support in a number of ways. Special thanks to my nephew *Animesh*, whose involvement gave a new breath to my research.

Many thanks to my comrades and fellow colleagues at SVIT, Vasad. It gives me a sense of happiness to be with you all. My humble acknowledgement to *The HOD ,EC department* and *The Chairman ,SVIT Vasad* for allowing me to pursue my research.

Finally, my heartfelt thanks to my wife *Prativa* and my children *Shruti* and *Siddhant* for their unconditional love and support. Words fail me to express my gratitude to my beloved parents and families of elder brothers who have sacrificed their comfort for my betterment.

I dedicate this piece of work to my late father who left for heavenly abode just few days before submission.

Sukant Kumar Chhotaray

Abstract

In most of the cryptological methods, the encrypted data or the cipher texts maintain same statistics of the plain texts, whereas matrix encryption method does not keep the statistics of individual cipher texts. However, it maintains the statistics of block of characters of size m where m is the size of the key matrix. One of the important features of the cipher matrix in Residue Number System (RNS) is that it is highly difficult and time consuming to obtain its inverse by standard inverse algorithms. Matrix in RNS does not have all the eigen values as defined in complex field. The eigen factors of a matrix is defined as the irreducible factors of the characteristic equation(eigen function). All the above properties are valid for cipher matrix in Galois Field. The public key is generated by using two types of matrices. One of these matrices is a self-invertible matrix or an orthonormal matrix in Galois field whereas the other matrix is a diagonally dominant matrix.

Matrix inversion is very difficult and time consuming when size of matrix and modulo number are large. The computational overhead in generalized Hill cipher can be reduced substantially by using self-invertible matrices. Self-invertible matrices uses less space compared to invertible matrices. In order to overcome this problem, p(modulo) is made very large so that there would be at least $p^{n/2}$ possible matrices making it extremely difficult for the intruder to find the key matrix. In this thesis several methods of generating self-invertible matrix are proposed.

Orthogonal Transform is used in signal processing. Modular Orthogonal Transform such as Walsh, Hadamard, Discrete Cosine Transform, Discrete Sine Transform, Discrete Fourier Transform have been used for encryption of image. The orthogonal matrices can be used as asymmetric key for encryption. In this work various methods of generating orthogonal matrices have been proposed. Matrix having primitive polynomial as eigen factors is used resulting in robust encryption.

A novel operation called exponentiation and its inverse has been defined in this thesis. All the properties of this new operation have been analyzed in Z_p. This operation is used for encryption of image. The original image can be obtained by

using the same exponentiation operation.

Chaotic sequence and chaotic signal generation is widely used in communication. Two stages of image encryption scheme using chaotic sequence is proposed in this work. First stage of encryption by chaotic sequence generated in $GF(p)$ and the second stge of encryption is carried out by one of the encryption methods discussed in the previous chapters.

Standard images have been used for encryption during simulation.

***Keywords*:** Encryption, Decryption, Cipher matrix, Public key, Private key, Residue number system, Eigen function, self-invertible matrix, Orthogonal, Galois Field, Exponentiation, Chaotic sequence.

Contents

List of Figures

List of symbols

- Z : set of integers. Example : $\{\ldots, -1, 0, 1, \ldots\}$
- Z_n : set of least residues modulo n. Example : $Z_6 = \{0, 1, 2, 3, 4, 5\}$
- Z_{n^*} : subset of Z_n, where each element has a multiplicative inverse.

Example : $Z_{6^*} = \{1, 5\}$

- Z_p : set of least residues modulo p, where p is prime.

Example : $Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$

- Z_{p^*} : subset of Z_p excluding 0. Example : $Z_{7^*} = \{1, 2, 3, 4, 5, 6\}$
- $G = \langle Z_n, + \rangle$: G is a commutative group where + operation is defined over Z_n
- $G = \langle Z_n, \times \rangle$: G is a commutative group where $\times$ operation is defined over Z_{n^*}
- $R = \langle Z, +, \times \rangle$: R is a commutative ring where two operations + and $\times$ are defined over Z and multiplicative inverse of the elements does not exists.
- $F = \langle Z, +, \times \rangle$: F is a commutative ring where multiplicative inverse of all the elements exists except of additive identity element. It is called as a field.
- GF(p) : Field defined over the set Z_p.
- GF(p^n) : Polynomial field where each term of the polynomial belongs to Z_n and all operations are defined modulo of an irreducible polynomial of degree n.

Chapter 1

Introduction

Chapter 1

Introduction

Cryptology is the branch of science that deals with the hiding of information and protection of important information from the intruder. In World War II, there was a need to secure the information on weapons, strategy and movement of military from the enemy. Presently in the era of information technology the security of information has become increasingly important. Since information is sent from sender to receiver through public communication channel, it is necessary to secure the information from other parties. Moreover, popular application of multimedia technology and increasing transmission ability of network gradually lead us to acquire information directly and clearly through images which should be protected from public. As e-governance is the present trend of administration and management, encryption of data has become a necessity. Image encryption has widespread applications including Government, military, financial institution, hospitals and private business.

1.1 Elementary Cryptosystems

A cryptosystem involves mapping of information from one domain to the same domain. The algorithm of mapping is called encryption and its inverse is called

decryption. The messages are enciphered by applying mathematical operations and the resulting messages are known as cipher texts. So the symbols that are encrypted will have the same kind of mathematical structure as the encrypted symbols. Since the number of symbols is finite, symbols must belong to finite group, ring or field. The algebraic manipulation of the symbols belonging to finite group, ring or field is used for encryption. Hence, it is also called algebraic cryptosystem. Specifically, the principle of linear algebra can be applied over the finite field, ring or group [1–4].

Cryptography can be broadly classified into symmetric (private-key/single-key) and asymmetric (public-key/two-key) cryptography. The symmetric cryptography involves the use of private-key encryption algorithm where the sender and the receiver share a closely related key. The decryption key corresponds to the inverse operation of the encryption key which can easily be formulated as there is no secrecy between the sender and the receiver in the cryptosystem. Asymmetric cryptography involves the use of two keys. One of the keys is a public-key, which may be known to anybody can be used to encrypt messages with signature. The private-key, known only to the recipient is used to decrypt messages and verify signatures. The inverse of the public key will be very difficult to obtain as some hidden parameters are only known to the recipient.

Encryption or information scrambling technology is an important security tool. Properly applied, it can provide a secure communication channel even when the underlying system and network infrastructure is not secure. This is particularly important when data passes through shared systems or network segments where more people may have access to the information. In these situations, sensitive data and especially passwords should be encrypted in order to protect it from unintended disclosure or modification. Encryption involves a mathematical trans-

formation of information into scrambled text, called "cipher text". The computational process (an algorithm) uses a key, actually just a big number associated with a password or pass phrase to compute or convert plain text into cipher text with numbers or strings of characters. The resulting encrypted text is decipherable only by the holder of the corresponding key. This deciphering process is called decryption.

E-governance and business transactions require several information security services. The information security services are confidentiality, integrity, availability, non-repudiation, authentication, etc. Confidentiality property is achieved through encryption. The authentication and integrity of message are normally achieved through biometric, digital signature and message authentication codes. Network security is another major aspect of security service. It consists of security in application layer, transport layer and the network layer [8].

1.2 Image Encryption

Rapid evolution of the internet in the digital world today has led to the security of digital images a very important feature attracting considerable attention in different image encryption methods. For example, medical diagnostic information in form of EEG, ECG, MRI, Sonograph of a particular patient have to be stored confidentially in the hospital. It is highly illegal to disclose the diagnostic data of a person to unauthorized person. There are various image encryption systems to encrypt and decrypt data. Due to large data size and real time constrains, algorithms that are good for textual data may not be suitable for multimedia data. In most of the natural images, the neighboring pixels are highly correlated. In order to dissipate the high correlation among pixels and increase the entropy, complex and efficient image encryption algorithm is necessary [10].

Protection of image data from unauthorized access is very important. Image encryption plays a significant role in the field of information hiding. Generally there are two levels of security for digital image encryption: low level and high level. In low level security encryption, the encrypted image has a degraded visual quality compared to that of the original one, but the content of the image is still visible and understandable to the viewers. In the high level security, the content is completely scrambled and the image appears as random noise. In such case, the visual characteristic of the image is not understandable to the viewers [11]. The proposed techniques of image encryption in this thesis can be categorised under high-level security encryption.

1.3 Background

In most of the cryptographic methods, the encrypted data or the cipher texts maintain same statistics of the plain texts, whereas matrix encryption method do not maintain the statistics of individual cipher texts. However, it maintains the statistics of block of characters of size equivalent to the size of the key matrix.

The method of substitution is one of the oldest techniques of encryption. In this technique, units of plaintext are replaced with ciphertext. The "units" may be single letters, pairs of letters, triplets of letters and mixtures of the above. A monoalphabetic substitution uses fixed substitution over the entire message, whereas a polyalphabetic substitution uses a number of substitutions at different positions in the message [10]. Several other algorithms based on permutation have been developed for encryption. The cipher matrix for encryption have been implemented in GF(p), GF(2^n), GF(p^n) and finite ring in Residue Number System modulo ($p_1 p_2$).

1.3.1 Galois Field

A ring is an algebraic structure defined over two operations. The first operation satisfies all the properties of an abelian group i.e. closure, associativity, commutativity, existence of an identity element, existence of an inverse element. But the second operation satisfies only the closure and associativity property [3]. A Galois Field is a ring where the second operation satisfies all the properties of an abelian group and the number of elements are finite. As the number of elements are finite, GF is a finite field. Galois fields are denoted by GF(p^n). Here p is a prime number. Depending upon the value of p and n Galois fields can be broadly divided into the following categories.

I. GF(p) : This field has maximum p elements. Example : $Z_p = 0, 1 \cdots p-1$.

II. GF(2^n) : This field has maximum 2^n elements. If set Z_p is used , $p \in [0, 2^n - 1]$ and p must be prime. Moreover, the value of n is usually 8, 16, 32 and so on. In this work, we have taken the value of n as 8.

III. GF(p^n) : This field has maximum p^n elements and p must be prime. In this work, we have taken the value of n as 2 and value of p as 13 [3].

1.3.2 Irreducible Polynomials

Polynomials are represented as n bit words where the coefficients are defined over GF(2). Here, finite field $GF(p)$ is not used. Rather the polynomial is represented as a n bit word for which the degree of the polynomial $\in [0, n-1]$. During multiplication of two polynomials, the degree of the resulting polynomial can become greater than $n-1$ which cannot be represented by the same field. For this reason, prime polynomials or irreducible polynomials are required. Degree of irreducible polynomial should be n so that when we divide the product of two polynomials by the irreducible polynomial, resulting polynomial can always

be defined over GF(2^n). Some examples of irreducible polynomials are $(x+1)$, x^2+x+1, x^3+x+1, x^4+x^3+1 and so on [3].

1.3.3 Residue Number System(RNS)

In Residue Number System (RNS) arithmetic, an integer z is uniquely represented by an n-tuple of integers $(x_1, x_2, \cdots, x_n)$, called the residue representation of z. The integers x_i, $i = 1, 2, \cdots, n$ are called the residues and are obtained as remainders when the number x is divided by a set of distinct and relatively prime integers, $m_i = 1, 2, \cdots, n$ called the moduli of the residue number system. Thus, $x_i = x_i \bmod (m_i)$, denoted by $|x_i|_{m_i}$, where $0 < x_i < m_i$ [6]. In RNS, arithmetic operations are performed concurrently on a number of smaller integers. The addition of two numbers x and y is given by: $x+y = (x_1, x_2, \cdots, x_n)+(y_1, y_2, \cdots, y_n) = (z_1, z_2, \cdots, z_n) = z$, where $z_i = |x_i + y_i|_{m_i}$. In a similar manner, multiplication is accomplished by taking the products of the corresponding residues modulo m_i. Thus, in each modulo channel, arithmetic can be performed on a pair of smaller integers thereby speeding up the whole operation. The actual speed depends on the number of bits used in each channel. In order to increase the speed the moduli must be kept small, but this in turn reduces the overall range of numbers that can be used. One of the important features of the cipher matrix in RNS is that it is very difficult and time consuming to obtain its inverse by standard inverse algorithms. In this work, one tuple RNS has been used.

1.4 Motivation

Remote sensing data (photographs taken by satellite) contain lot of information regarding natural resources like mines and agriculture. These photographs or images should be encrypted and protected from the intruder. There has been

a lot of research on image compression and image encryption. However, very little research has been done on image encryption using cipher matrix based on symmetric or asymmetric keys. Here it is intended to introduce a novel method of cipher matrix encryption utilizing the asymmetric key concept. Moreover, the method of encryption is made more robust by using two stage encryption. In the first stage, each pixel of the image is modified using integer chaotic sequence and in the second stage the novel asymmetric key encryption technique is used to encrypt the modified image.

1.5 Objective of the Thesis

The objective of present research work is to investigate on different techniques of encryption in asymmetric cryptosystems. In summary, the main objectives of the research work can be listed below.

- To generate self-invertible cipher matrix in GF(p), GF(2^8) and GF(p^n).
- To develop Orthogonal cipher matrix for image encryption in GF(p), GF(2^8) and GF(p^n).
- To introduce Exponentiation operation in GF(p) for encryption of image.
- A novel concept of Image Encryption by Chaotic sequence in GF(p).

1.6 Thesis Layout

Rest of the thesis is organised as follows-

Chapter 2: Self-invertible matrix and Image Encryption Inversion of a matrix is extremely difficult and time consuming when size of matrix and modulo number are large [12–15, 22]. The computational overhead in generalized Hill cipher can be reduced substantially by using self-invertible matrices. Using self-invertible matrices instead of invertible matrices decreases the key space. In order to overcome this problem, p(modulo) is made very large so that there would be at least $p^{n/2}$ possible matrices making it nearly impossible for the intruder to find the key matrix. Here, several methods of generating self-invertible matrix are presented. In order to make encryption more robust, another matrix B is multiplied with A to generate the public key matrix.

Chapter 3: Discrete Orthogonal Transform and Image Encryption Orthogonal Transform is a very popular technique in signal processing. Modular Orthogonal Transform such as Walsh, Hadamard, Discrete Cosine Transform, Discrete Sine Transform, Discrete Fourier Transform have been used for encryption of image [11,17,28–32,39,42,43]. The orthogonal matrices can be used as asymmetric key (similar to self-invertible matrices discussed earlier) for encryption. Several methods for generating orthogonal matrices have been proposed. A matrix having primitive polynomial as eigen factors is used to make the encryption more robust.

Chapter 4: Use of Exponentiation to Encrypt an Image A novel operation called exponentiation has been defined as

$$A * *B = C$$

where $C_{ij} = \prod_{k=1}^{n} a_{ik} * *b_{kj}$ for i = 1,$\cdots$,n and j = 1,$\cdots$,m.
The exponentiation inverse of matrix has also been defined.

All the properties of this new operation have been analyzed in Z_p. This operation

is used for encryption of image. If B is $m \times m$ key matrix and A = $m \times 1$ image pixel vector, then cipher text C is obtained as $A**B$ = C. It is shown that obtaining the original image requires the same exponentiation operation.

Chapter 5: Image Encryption by Integer Chaotic Sequence Large number of researchers have used chaotic sequence and chaotic signal generation in communication [23–27]. In this chapter, we have introduced two stages of encryption. First encryption is done using chaotic sequence followed by any one of the encryption methods as discussed in the previous chapters. The proposed scheme utilizes the chaotic sequence generated in $GF(p)$.
Chaotic sequence y in GF(p) can be generated as

$$x(i) = x(i-1) \times \mu \times (1 - x(i-1))$$

$$z_i = (x(i) * Q)$$

$$z = round(z_i * N)$$

$$y(i) = mod(z, p)$$

where $1 \leq i \leq 256$ and initial value of x i.e. $x(1) = 0.2$, μ=3.8, Q=105, $N = 10^5$ and $p = 251$. Since different chaotic sequence can be generated by varying initial value $x(1)$[0,1], μ [3.6,4) and Q[104,106] these can be considered as private keys.

Chapter 6: Conclusion and Future Work This chapter reports overall contributions of the thesis. Different methods of generating self-invertible matrix have been proposed. In order to make encryption robust, a sparse matrix is defined whose inverse can be obtained easily. The self-invertible matrix and sparse matrix are used as private keys in the asymmetric key cryptosystem. Several methods of generating orthonormal matrices has been proposed and used in combination with

the sparse matrix for encryption of images. Exponentiation operation on matrices is introduced and used for image encryption. Finally, a two-stage encryption technique based on chaotic sequence followed by any one of the cipher matrix technique proposed earlier is used to make encryption more robust. Also future research problems are outlined for further investigation on the same/related topics which include:

• Generating self-invertible and othonormal matrices over Z_n and using them for encryption.

• Application of chinese remainder theorem in GF(p^n) for encryption.

• Encryption by representing data by variable radix number system and quantum number system.

• Introduction of error correcting codes for making cryptography more secured and immune to channel noise.

Chapter 2

Self Invertible Matrix and Image Encryption

Chapter 2

Self-Invertible Matrix and Image Encryption

A poly-alphabetic cipher is a block cipher where the plain text character is encrypted in such a way that the corresponding cipher text character will not be same each time. Hill cipher is a poly-alphabetic cipher developed by Lester Hill and the method of encryption is one of the oldest methods of encryption. In this method, the message is first converted into blocks of equal size (say n) which constitutes the matrix of the plain text. A key matrix of order $n \times n$ is chosen such that its inverse exists. The cipher text matrix is obtained by multiplying the plain text matrix with the key matrix. If Hill cipher is used for encryption, the intruder has to test maximum of p^{n^2} combination of characters in order to guess the key matrix correctly by using brute force attack. Here, p represents the size of the domain of characters. If we consider only the set of lower case alphabets, then $p = 26$ and the maximum number of trials will be 26^{n^2}. In generalized Hill cipher, the key matrix is a combination of a $n \times n$ matrix and a vector of length n. The inclusion of the vector in the key matrix results in a higher level of security than the original Hill cipher encryption method. This is due to the fact that the brute

force attack becomes extremely difficult, as the intruder has to test maximum of p^{n^2+n} characters instead of p^{n^2} characters. The increase in security is quite substantial which is evident from the following example. If $n = 5$ and generalized Hill cipher is used for encryption, the increase in number of trials to guess the key matrix will be $26^5 = 1, 18, 81, 376$. Hence, an extension of generalized Hill cipher method is proposed that yields a higher level of security than the generalized Hill cipher method [5].

2.1 Proposed methods of generating self-invertible matrices in GF(p)

The computational overhead in generalized Hill cipher can be reduced substantially by using self-invertible matrices. A matrix D is self-invertible if $D^2 = I$ or $D = D^{-1}$. Using self-invertible matrices instead of invertible matrices decreases the key space as the number of self-invertible matrices of a particular order say $n \times n$ is very less compared to the number of invertible matrices of the same order. In order to overcome this problem, p(modulo) is made very large so that there would be at least $p^{n/2}$ possible matrices making it nearly impossible for the intruder to find the key matrix. Several methods of generating self-invertible matrices are presented here. In this analysis, l, m, n are chosen to be integers.

2.1.1 Generation of self-invertible matrix from another self-invertible matrix

a) Method 1

Let A be a $(l + m + n) \times (l + m + n)$ self-invertible matrix expressed in terms

of partition matrices

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{2.1}$$

where $A_{11}, A_{12}, A_{13}, A_{21}, A_{22}, A_{23}, A_{31}, A_{32}$ and A_{33} are the partition matrices of size $(l \times l), (l \times m), (l \times n), (m \times l), (m \times m), (m \times n), (n \times l), (n \times m)$ and $(n \times n)$ respectively.

Since A is self-invertible, $A = A^{-1}$

i.e. $A.A = I$, which can be written as

$$A^2 = \begin{bmatrix} A_{11}A_{11} + A_{12}A_{21} + A_{13}A_{31} & A_{11}A_{12} + A_{12}A_{22} + A_{13}A_{32} & A_{11}A_{13} + A_{12}A_{23} + A_{13}A_{33} \\ A_{21}A_{11} + A_{22}A_{21} + A_{23}A_{31} & A_{21}A_{12} + A_{22}A_{22} + A_{23}A_{32} & A_{21}A_{13} + A_{22}A_{23} + A_{23}A_{33} \\ A_{31}A_{11} + A_{32}A_{21} + A_{33}A_{31} & A_{31}A_{12} + A_{32}A_{22} + A_{33}A_{32} & A_{31}A_{13} + A_{32}A_{23} + A_{33}A_{33} \end{bmatrix}$$

$$= \begin{bmatrix} I_{11} & 0 & 0 \\ 0 & I_{22} & 0 \\ 0 & 0 & I_{33} \end{bmatrix} \tag{2.2}$$

Here, I is an identity matrix of order $(l + m + n) \times (l + m + n)$ and hence can be represented in terms of partition matrices similar to A.

If a matrix B is generated by multiplying A_{12} and A_{32} by k (any integer) and dividing A_{21} and A_{23} by k, then the matrix B also becomes self-invertible.

Proof- As per the above procedure,

$$B = \begin{bmatrix} A_{11} & kA_{12} & A_{13} \\ \frac{1}{k}A_{21} & A_{22} & \frac{1}{k}A_{23} \\ A_{31} & kA_{32} & A_{33} \end{bmatrix} \tag{2.3}$$

Then,

$$B^2 = \begin{bmatrix} A_{11}A_{11} + A_{12}A_{21} + A_{13}A_{31} & k(A_{11}A_{12} + A_{12}A_{22} + A_{13}A_{32}) & A_{11}A_{13} + A_{12}A_{23} + A_{13}A_{33} \\ \frac{1}{k}(A_{21}A_{11} + A_{22}A_{21} + A_{23}A_{31}) & A_{21}A_{12} + A_{22}A_{22} + A_{23}A_{32} & \frac{1}{k}(A_{21}A_{13} + A_{22}A_{23} + A_{23}A_{33}) \\ A_{31}A_{11} + A_{32}A_{21} + A_{33}A_{31} & k(A_{31}A_{12} + A_{32}A_{22} + A_{33}A_{32}) & A_{31}A_{13} + A_{32}A_{23} + A_{33}A_{33} \end{bmatrix}$$

Using (2.2) in above equation

$$B^2 = \begin{bmatrix} I_{11} & 0 & 0 \\ 0 & I_{22} & 0 \\ 0 & 0 & I_{33} \end{bmatrix} \tag{2.4}$$

or $B^2 = I$

Example (Modulo 13) :

$$\text{Let } A = \begin{bmatrix} 8 & 10 & 4 & 10 & 2 \\ 2 & 1 & 8 & 6 & 9 \\ 8 & 7 & 11 & 2 & 3 \\ 3 & 11 & 9 & 7 & 9 \\ 12 & 3 & 7 & 8 & 0 \end{bmatrix}$$

The matrix A consists of the following partition matrices.

$$A_{11} = \begin{bmatrix} 8 & 10 \\ 2 & 1 \end{bmatrix},\ A_{12} = \begin{bmatrix} 4 & 10 \\ 8 & 6 \end{bmatrix},\ A_{13} = \begin{bmatrix} 2 \\ 9 \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} 8 & 7 \\ 3 & 11 \end{bmatrix},\ A_{22} = \begin{bmatrix} 11 & 2 \\ 9 & 7 \end{bmatrix},\ A_{23} = \begin{bmatrix} 3 \\ 9 \end{bmatrix}$$

$$A_{31} = \begin{bmatrix} 12 \\ 3 \end{bmatrix},\ A_{32} = \begin{bmatrix} 7 \\ 8 \end{bmatrix},\ A_{33} = \begin{bmatrix} 0 \end{bmatrix}$$

For $k = 2$

$$B = \begin{bmatrix} 8 & 10 & 8 & 7 & 2 \\ 2 & 1 & 3 & 12 & 9 \\ 4 & 10 & 11 & 2 & 8 \\ 8 & 12 & 9 & 7 & 11 \\ 12 & 3 & 1 & 3 & 0 \end{bmatrix}$$

$$B^2 \bmod(13) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, B is self-invertible in modulo(13).

b) Method 2

Let A be a $(m+n) \times (m+n)$ self-invertible matrix expressed in terms of partition matrices.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{2.5}$$

where A_{11}, A_{12}, A_{21} and A_{22} are the partition matrices of size $(m \times m), (m \times n), (n \times m)$ and $(n \times n)$.

$$\text{and } B = \begin{bmatrix} -A_{11} & A_{12} \\ A_{21} & -A_{22} \end{bmatrix} \tag{2.6}$$

then B becomes self-invertible.

Proof of above method is self evident.

Example (Modulo 13) :

$$\text{Let } A = \begin{bmatrix} 5 & 3 & 8 & 7 & 2 \\ 11 & 12 & 3 & 12 & 9 \\ 4 & 10 & 2 & 11 & 5 \\ 8 & 12 & 4 & 6 & 2 \\ 12 & 3 & 12 & 10 & 0 \end{bmatrix}$$

Let $m = 3$ and $n = 2$. Then the matrix A consists of the following partition matrices.

$$A_{11} = \begin{bmatrix} 5 & 3 & 8 \\ 11 & 12 & 3 \\ 4 & 10 & 2 \end{bmatrix}, A_{12} = \begin{bmatrix} 7 & 2 \\ 12 & 9 \\ 11 & 5 \end{bmatrix}$$
$$A_{21} = \begin{bmatrix} 8 & 12 & 4 \\ 12 & 3 & 12 \end{bmatrix}, A_{22} = \begin{bmatrix} 6 & 2 \\ 10 & 0 \end{bmatrix}$$

$$\text{Now } B = \begin{bmatrix} -A_{11} & A_{12} \\ A_{21} & -A_{22} \end{bmatrix}$$

Substituting the values of A_{11}, A_{12}, A_{21} and A_{22} in the above matrix,

$$B = \begin{bmatrix} -5 & -3 & -8 & 7 & 2 \\ -11 & -12 & -3 & 12 & 9 \\ -4 & -10 & -2 & 11 & 5 \\ 8 & 12 & 4 & -6 & -2 \\ 12 & 3 & 12 & -10 & 0 \end{bmatrix}$$

$$B \bmod(13) = \begin{bmatrix} 8 & 10 & 5 & 7 & 2 \\ 2 & 1 & 10 & 12 & 9 \\ 9 & 3 & 11 & 11 & 5 \\ 8 & 12 & 4 & 7 & 11 \\ 12 & 3 & 12 & 3 & 0 \end{bmatrix}$$

$$B^2 \bmod(13) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, B is self-invertible in modulo(13).

c) Method 3

Let A and B be self-invertible matrices of size $(m \times m)$ and $(n \times n)$ respectively. Then C will be self-invertible if

$$C = \begin{bmatrix} a_{11} \times B & a_{12} \times B & \dots & a_{1m} \times B \\ a_{21} \times B & a_{22} \times B & \dots & a_{2m} \times B \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} \times B & a_{m2} \times B & \dots & a_{mm} \times B \end{bmatrix} \tag{2.7}$$

or

$$C = \begin{bmatrix} b_{11} \times A & b_{12} \times A & \dots & b_{1n} \times A \\ b_{21} \times A & b_{22} \times A & \dots & b_{2n} \times A \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} \times A & b_{n2} \times A & \dots & b_{nn} \times A \end{bmatrix} \tag{2.8}$$

Proof - Case 1 (equation (2.7)):

$$C^2 = \begin{bmatrix} ({a_{11}}^2 + a_{12}a_{21} + \cdots + a_{1m}a_{m1})B^2 & \dots & (a_{11}a_{1m} + a_{12}a_{2m} + \cdots + a_{1m}a_{mm})B^2 \\ \cdots\cdots\cdots\cdots\cdots\cdots & \dots & \cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots & \dots & \cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots & \dots & \cdots\cdots\cdots\cdots\cdots\cdots \\ (a_{m1}a_{11} + a_{m2}a_{21} + \cdots + a_{mm}a_{m1})B^2 & \dots & (a_{m1}a_{1m} + a_{m2}a_{2m} + \cdots + a_{mm}^2)B^2 \end{bmatrix}$$

$$= \begin{bmatrix} B^2 & 0 & 0 & \cdots & 0 \\ 0 & B^2 & 0 & \cdots & 0 \\ 0 & 0 & B^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & B^2 \end{bmatrix}$$

$$= I$$

Hence, C is self-invertible.

Proof - Case 2 (equation (2.8)):

Similarly, C generated by equation (2.8) can be proved to be self-invertible.

d) Method 4

Let A be a matrix of size $(m \times m)$ and

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \tag{2.9}$$

if $B_{11} = A^3$, $B_{22} = -A^3$ and $B_{12} \times B_{21} = I - A^6$

then, B will be self-invertible. Therefore, B_{12} can be any factor of the expression $I - A^6$. By using above method, we can get several self-invertible matrices. The proof is self evident.

Example (Modulo 13) :

$$\text{Let } A = \begin{bmatrix} 3 & 6 & 5 \\ 2 & 9 & 10 \\ 4 & 7 & 11 \end{bmatrix} \text{ so } A^3 = \begin{bmatrix} 12 & 0 & 1 \\ 6 & 6 & 9 \\ 4 & 6 & 8 \end{bmatrix}$$

Case I -

$$\text{Let } B_{12} = I, \text{ then } B_{21} = I - A^6 = \begin{bmatrix} 9 & 7 & 6 \\ 12 & 2 & 11 \\ 1 & 7 & 9 \end{bmatrix}, \text{so } B = \begin{bmatrix} 12 & 0 & 1 & 1 & 0 & 0 \\ 6 & 6 & 9 & 0 & 1 & 0 \\ 4 & 6 & 8 & 0 & 0 & 1 \\ 9 & 7 & 6 & 1 & 0 & 12 \\ 12 & 2 & 11 & 7 & 7 & 4 \\ 1 & 7 & 9 & 9 & 7 & 5 \end{bmatrix}$$

Case II -

$$\text{Let } B_{12} = I - A, \text{ then } B_{21} = (I + A)(I + A^2 + A^4), \text{so } B = \begin{bmatrix} 12 & 0 & 1 & 11 & 7 & 8 \\ 6 & 6 & 9 & 11 & 5 & 3 \\ 4 & 6 & 8 & 9 & 6 & 3 \\ 9 & 2 & 2 & 1 & 0 & 12 \\ 7 & 9 & 2 & 7 & 7 & 4 \\ 7 & 0 & 6 & 9 & 7 & 5 \end{bmatrix}$$

Case III -

$$\text{Let } B_{12} = I + A, \text{ then } B_{21} = (I - A)(I + A^2 + A^4), \text{so } B = \begin{bmatrix} 12 & 0 & 1 & 4 & 6 & 5 \\ 6 & 6 & 9 & 2 & 10 & 10 \\ 4 & 6 & 8 & 4 & 7 & 12 \\ 12 & 6 & 10 & 1 & 0 & 12 \\ 6 & 1 & 3 & 7 & 7 & 4 \\ 11 & 11 & 0 & 9 & 7 & 5 \end{bmatrix}$$

Case IV -

$$\text{Let } B_{12} = I - A^2, \text{ then } B_{21} = (I + A^2 + A^4), \text{so } B = \begin{bmatrix} 12 & 0 & 1 & 12 & 10 & 0 \\ 6 & 6 & 9 & 1 & 7 & 11 \\ 4 & 6 & 8 & 8 & 5 & 11 \\ 4 & 4 & 6 & 1 & 0 & 12 \\ 0 & 5 & 9 & 7 & 7 & 4 \\ 9 & 12 & 3 & 9 & 7 & 5 \end{bmatrix}$$

Case V -

$$\text{Let } B_{12} = I - A^3, \text{ then } B_{21} = (I + A^3), \text{so } B = \begin{bmatrix} 12 & 0 & 1 & 2 & 0 & 12 \\ 6 & 6 & 9 & 7 & 8 & 4 \\ 4 & 6 & 8 & 9 & 7 & 6 \\ 0 & 0 & 1 & 1 & 0 & 12 \\ 6 & 7 & 9 & 7 & 7 & 4 \\ 4 & 6 & 9 & 9 & 7 & 5 \end{bmatrix}$$

Case VI -

$$\text{Let } B_{12} = (I - A)(I - A + A^2), \text{ then } B_{21} = (I + A + A^2), \text{so } B = \begin{bmatrix} 12 & 0 & 1 & 0 & 7 & 2 \\ 6 & 6 & 9 & 1 & 4 & 1 \\ 4 & 6 & 8 & 11 & 9 & 3 \\ 10 & 5 & 11 & 1 & 0 & 12 \\ 8 & 0 & 7 & 7 & 7 & 4 \\ 9 & 10 & 11 & 9 & 7 & 5 \end{bmatrix}$$

All other cases are transpose of all the matrices cited above. This method can also be extended by taking any value of n, i.e. by assuming $B_{11} = A^n$, for $n > 3$.

e) Method 5

I. If A is a self-invertible matrix and B is defined as below for any value of m and n such that modulo square root of $m^2 + n^2$ exists.

$$B = \frac{1}{\sqrt{n^2 + m^2}} \begin{bmatrix} nA & mA \\ mA & -nA \end{bmatrix} \tag{2.10}$$

then B is self-invertible.

Example (Modulo 13) :

$$\text{Let } A = \begin{bmatrix} 6 & 6 & 7 \\ 9 & 4 & 10 \\ 4 & 10 & 4 \end{bmatrix}$$

putting n=6 and m=2 in (2.10)

$$B = \begin{bmatrix} 10 & 10 & 3 & 12 & 12 & 1 \\ 2 & 11 & 8 & 5 & 8 & 7 \\ 11 & 8 & 11 & 8 & 7 & 8 \\ 12 & 12 & 1 & 3 & 3 & 10 \\ 5 & 8 & 7 & 11 & 2 & 5 \\ 8 & 7 & 8 & 2 & 5 & 2 \end{bmatrix}$$

which is self-invertible.

II. If A is a self-invertible matrix, then for any integer values of $m_1, m_2, ..., m_6$;

$$B = \frac{1}{\sqrt{m_1^2 + m_2^2 + m_3 m_5 + m_4 m_6}} \begin{bmatrix} m_1 A + m_2 I & m_3 A + m_4 I \\ m_5 A + m_6 I & -(m_1 A + m_2 I) \end{bmatrix} \tag{2.11}$$

B will be a self-invertible matrix provided

$$2m_1 m_2 + m_4 m_5 + m_3 m_6 = 0 \tag{2.12}$$

and modulo square root of $m_1^2 + m_2^2 + m_3m_5 + m_4m_6$ exists.

One of the solutions of the above equation is

$$m_1 = 3,\ m_2 = 4,\ m_3 = 5,\ m_4 = 6,\ m_5 = 7, \text{ and } m_6 = 5$$

The above solution satisfies equation (2.12) as

$$2m_1m_2 + m_4m_5 + m_3m_6 = 24 + 42 + 25 = 91\text{mod}(13) = 0$$

$$m_1^2 + m_2^2 + m_3m_5 + m_4m_6 = 9 + 16 + 35 + 30 = 90\,\text{mod}13 = -1\,\text{mod}13$$

$$\text{and } \sqrt{-1}\,\text{mod}(13) = 5 \text{ or } 8$$

Example (Modulo 13) :

$$Let\ A = \begin{bmatrix} 6 & 6 & 7 \\ 9 & 4 & 10 \\ 4 & 10 & 4 \end{bmatrix}$$

substituting modulo square root of $m_1^2 + m_2^2 + m_3m_5 + m_4m_6 = 8$ in equation (2.11)

$$B = \frac{1}{8}\begin{bmatrix} 9 & 5 & 8 & 10 & 4 & 9 \\ 1 & 3 & 4 & 6 & 0 & 11 \\ 12 & 4 & 3 & 7 & 11 & 0 \\ 8 & 3 & 10 & 4 & 8 & 5 \\ 11 & 7 & 5 & 12 & 10 & 9 \\ 2 & 5 & 7 & 1 & 9 & 10 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 12 & 1 & 11 & 7 & 6 \\ 5 & 2 & 7 & 4 & 0 & 3 \\ 8 & 7 & 2 & 9 & 3 & 0 \\ 1 & 2 & 11 & 7 & 1 & 12 \\ 3 & 9 & 12 & 8 & 11 & 6 \\ 10 & 12 & 9 & 5 & 6 & 11 \end{bmatrix}$$

This resulting matrix is self-invertible.

Again substituting modulo square root of $m_1^2 + m_2^2 + m_3m_5 + m_4m_6 = 5$ in equation(2.11)

$$B = \frac{1}{5}\begin{bmatrix} 9 & 5 & 8 & 10 & 4 & 9 \\ 1 & 3 & 4 & 6 & 0 & 11 \\ 12 & 4 & 3 & 7 & 11 & 0 \\ 8 & 3 & 10 & 4 & 8 & 5 \\ 11 & 7 & 5 & 12 & 10 & 9 \\ 2 & 5 & 7 & 1 & 9 & 10 \end{bmatrix}$$

$$= \begin{bmatrix} 7 & 1 & 12 & 2 & 6 & 7 \\ 8 & 11 & 6 & 9 & 0 & 10 \\ 5 & 6 & 11 & 4 & 10 & 0 \\ 12 & 11 & 2 & 6 & 12 & 1 \\ 10 & 4 & 1 & 5 & 2 & 7 \\ 3 & 1 & 4 & 8 & 7 & 2 \end{bmatrix}$$

This matrix is also self-invertible.

f) Method 6

If A is a matrix such that $A \times A = 0$, then for any integer values of $m_1, m_2, ..., m_6$;

$$B = \frac{1}{\sqrt{m_2^2 + m_4m_6}}\begin{bmatrix} m_1A + m_2I & m_3A + m_4I \\ m_5A + m_6I & -(m_1A + m_2I) \end{bmatrix} \tag{2.13}$$

B will be a self-invertible matrix provided

$$2m_1m_2 + m_4m_5 + m_3m_6 = 0 \tag{2.14}$$

and $m_2^2 + m_4m_6$ is a perfect square. One of the solutions of the above equation is:

$$m_1 = 2,\ m_2 = 6,\ m_3 = 5,\ m_4 = 6,\ m_5 = 7 \text{ and } m_6 = 5$$

This solution satisfies equation (2.14) as

$$2m_1m_2 + m_4m_5 + m_3m_6 = 24 + 42 + 25 = 91\text{mod}(13) = 0$$

and

$$m_2^2 + m_4 m_6 = 36 + 30 = 66 = 1\text{mod}(13) \text{ and } \sqrt{1} = 1$$

Example (Modulo 13) :

$$\text{Let } A = \begin{bmatrix} 6 & 12 & 11 \\ 1 & 2 & 4 \\ 11 & 9 & 5 \end{bmatrix}$$

then $A \times A = 0$

Substituting $m_1 = 2, m_2 = 6, m_3 = 5, m_4 = 6, m_5 = 7, m_6 = 5$ and $\sqrt{m_2^2 + m_4 m_6} = 1$.

$$B = \begin{bmatrix} 5 & 11 & 9 & 10 & 8 & 3 \\ 2 & 10 & 8 & 5 & 3 & 7 \\ 9 & 5 & 3 & 3 & 6 & 5 \\ 8 & 6 & 12 & 8 & 2 & 4 \\ 7 & 6 & 2 & 11 & 3 & 5 \\ 12 & 11 & 1 & 4 & 8 & 10 \end{bmatrix}$$

This matrix is self-invertible.

2.1.2 Generation of self-invertible matrix from a random matrix in GF(p)

If A is a $m \times m$ self-invertible matrix partitioned as $A_{11}(1 \times 1)$, $A_{12}(1 \times (m-1))$, $A_{21}((m-1) \times 1)$ and $A_{22}((m-1) \times (m-1))$, then it can be generated from a random matrix B by the following method.

1. Generate a random matrix B of size $(m-1)\times(m-1)$ with $b_{ij} = i \times s \times r^{j-1} mod(p)$ where r and s are any numbers in GF(p). Since all the row vectors of B are linearly dependent, all eigen values except one will be zero and the non-zero eigen value will be equal to the trace of the matrix.

2. Generate A_{22} by adding I (Identity matrix) to B, or $A_{22} = B + I$

3. A_{22} will have one eigen value $\lambda_1 = trace(A_{22}) - m + 2$ and all other eigen values

will be 1.

4. Let $A_{11} = -\lambda_1$

5. Then, A_{12} and A_{21} are consistent solutions of $I - {A_{22}}^2$ i.e. $A_{21}.A_{12} = I - {A_{22}}^2$.

6. If A is generated in the following way, it will be self-invertible.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

Proof:

If A is a self-invertible matrix, then

$$A_{12}.A_{21} = 1 - {\lambda_1}^2 \tag{2.15}$$

$$A_{21}.A_{12} = I - {A_{22}}^2 \tag{2.16}$$

$$A_{12}(-\lambda_1 I + A_{22}) = 0 \tag{2.17}$$

$$\text{and } (-\lambda_1 I + A_{22})A_{21} = 0 \tag{2.18}$$

A_{12} and A_{21} are consistent solutions of $I - {A_{22}}^2$ as stated above which satisfies (2.16).

Moreover,

$$A_{12} \times A_{21} = trace\ of\ \{I - {A_{22}}^2\}$$

Therefore,

$$A_{12}.A_{21} = 1 - {\lambda_1}^2$$

If both (2.15) and (2.16) are satisfied, then (2.17) and (2.18) will be automatically satisfied as ${A_{12}}^T$ and A_{21} are left and right eigen vectors of A_{22}. Thus, it is proved that A is self-invertible.

Example (Modulo 13) :

Let B be a 4×4 partition matrix with $s = 2$ and $r = 6$

$$B = \begin{bmatrix} 2 & 12 & 7 & 3 \\ 4 & 11 & 1 & 6 \\ 6 & 10 & 8 & 9 \\ 8 & 9 & 2 & 12 \end{bmatrix}$$

Since $A_{11} = -\lambda_1 = -8 = 5$

$$A_{22} = B + I = \begin{bmatrix} 3 & 12 & 7 & 3 \\ 4 & 12 & 1 & 6 \\ 6 & 10 & 9 & 9 \\ 8 & 9 & 2 & 0 \end{bmatrix}$$

$$I - {A_{22}}^2 = \begin{bmatrix} 8 & 9 & 2 & 12 \\ 3 & 5 & 4 & 11 \\ 11 & 1 & 6 & 10 \\ 6 & 10 & 8 & 9 \end{bmatrix}$$

On solving (2.15) and (2.16), we get one of the consistent solution A_{21} and A_{12}, whose values are $\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}^T$ and $\begin{bmatrix} 8 & 9 & 2 & 12 \end{bmatrix}$ respectively. Thus, the self-invertible matrix will be

$$A = \begin{bmatrix} 5 & 8 & 9 & 2 & 12 \\ 1 & 3 & 12 & 7 & 3 \\ 2 & 4 & 12 & 1 & 6 \\ 3 & 6 & 10 & 9 & 9 \\ 4 & 8 & 9 & 2 & 0 \end{bmatrix}$$

2.2 Generation of self-invertible matrix in GF(p^n)

2.2.1 Generation of 2 × 2 self-invertible matrix

If $q(x)$ is a primitive polynomial of degree n, and matrix A is defined as

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{2.19}$$

then A will be self-invertible provided that

$$a_{12}(x)[a_{11}(x) + a_{22}(x)] = 0 \tag{2.20}$$

$$\text{and} \quad a_{11}^2(x) + (a_{12}(x)a_{21}(x)) = 1 \tag{2.21}$$

Equations (2.20) and (2.21) imply that

$$a_{11}(x) = -a_{22}(x) \tag{2.22}$$

$$\text{and } a_{21}(x) = (1 - a_{11}^2(x))(a_{12}^{-1}(x)) \tag{2.23}$$

2.2.2 Generation of $m \times m$ self-invertible matrix

Let A be a $m \times m$ matrix mod $q(x)$, where $q(x)$ is the primitive polynomial in GF (p^n).

If $B_{11} = A^3$, $B_{12} = I + A^2$, $B_{21} = I + A^2 + A^4$ and $B_{22} = -B_{11}$

$$\text{then } B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & -B_{11} \end{bmatrix} \tag{2.24}$$

Here, B is a self-invertible matrix mod $q(x)$.

The methods enumerated in the previous section 2.1.2 for obtaining self-invertible matrix in GF(p) are also applicable for generation of self-invertible matrix in GF (p^n).

So,

$$B = \frac{1}{\sqrt{n^2(x) + m^2(x)}} \begin{bmatrix} n(x)A(x) & m(x)A(x) \\ -m(x)A(x) & n(x)A(x) \end{bmatrix}$$

provided $m^2(x) + n^2(x)$ equals to a complete square in GF(p). This can be generated by the following relation as

$$m = \frac{m_1^2(x) - m_2^2(x)}{m_1^2(x) + m_2^2(x)} \times K \tag{2.25}$$

$$n = \frac{2m_1(x)m_2(x)}{m_1^2(x) + m_2^2(x)} \times K \tag{2.26}$$

where K is any number.

2.3 Eigen value problem of matrices in GF(p)

In section 2.1.2, generation of higher order self-invertible matrices depend on eigen values of lower order matrices. The conventional definition of eigen value

of matrices in GF(p) is just not sufficient as the characteristic equation of matrix does not always possess roots in GF(p) [20–22]. The characteristic equation can be factorized (if not irreducible) into a set of irreducible polynomial factors. We call these factors as eigen factors of a matrix. The conventional eigen vectors of matrix in GF(p) can be defined as long as the eigen factor of the matrix is linear. Therefore, the modal matrix of the original matrix can only be defined provided that all the eigen factors of the matrix are linear. If $\lambda^n + a_{n-1}\lambda^{(n-1)} + \cdots + a_0$ is a characteristic equation(eigen function) of a matrix, one of the corresponding matrices in GF(p) will be

$$\begin{bmatrix} -a_{n-1} & 1 & 0 & 0 & 0 & \cdots & 0 \\ -a_{n-2} & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ -a_1 & 0 & 0 & 0 & 0 & \cdots & 1 \\ -a_0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

or

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & -a_4 & \vdots & -a_{n-1} \end{bmatrix}$$

2.3.1 Proposed theorem on eigen factor of a matrix in GF(p)

THEOREM 2.1

If eigen function of an $r \times r$ matrix A is a primitive polynomial in Z_p then $A^n = I$, where n is the smallest integer that equals to $p^r - 1$.

Proof - Since $r \times r$ matrix A is defined over Z_p, the eigen factor belongs to GF(p^r). As per the definition of primitive polynomial if 'a' is a primitive polynomial then $a^n = 1$. By Sylvster theorem the eigen function satisfies the matrix equation and

therefore, the primitive polynomial will be replaced by the matrix A. Therefore $A^n = I$

Corollary- If the eigen function equals to the product of two primitive polynomials of degree r_1 and r_2, then the minimum integer n can be defined as below.

$$n = (p^{r_1} - 1)(p^{r_2} - 1) \tag{2.27}$$

Since $A^n = I$ in Z_p, one can obtain the inverse of a matrix by determining A^{n-1}. Thus, the inversion process becomes easier as one can obtain the inverse of a matrix A by multiplying A with itself $(n-1)$ times.

2.3.2 Proposed method to obtain inverse of a matrix

If B is a matrix generated by the following method

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

where $B_{12}, B_{13}, B_{21}, B_{23}, B_{31}andB_{32}$ are matrices with all zero elements and B_{11}, B_{22}, B_{33} may be either diagonal matrices or matrices similar to matrices defined in section 2.3.

$$\text{then } B^{-1} = \begin{bmatrix} B_{11}^{-1} & B_{12} & B_{13} \\ B_{21} & B_{22}^{-1} & B_{23} \\ B_{31} & B_{32} & B_{33}^{-1} \end{bmatrix}$$

The inverse of a diagonal matrix is also diagonal with elements equal to the inverse of the diagonal elements of the matrix. Therefore. decryption matrix can be easily obtained.

The inverse of matrices defined in section 2.3 can be found out by the following method.

Let A be a matrix of the form defined in section 2.3 and B be its inverse i.e

$B = A^{-1}$

$$A = \begin{bmatrix} -a_{n-1} & 1 & 0 & 0 & 0 & \cdots & 0 \\ -a_{n-2} & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ -a_1 & 0 & 0 & 0 & 0 & \cdots & 1 \\ -a_0 & 0 & 0 & 0 & 0 & \vdots & 0 \end{bmatrix} \tag{2.28}$$

$$B = \begin{bmatrix} 0 & 0 & \cdots & 0 & b_{n-1} \\ 1 & 0 & \cdots & 0 & \cdot \\ 0 & 1 & \cdots & \cdots & \cdot \\ \cdots & \cdots & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & \cdots & b_0 \end{bmatrix} \tag{2.29}$$

Since, $B = A^{-1} \implies AB = I$ and using this relation, the elements of B can be found out in the following way.

$$-a_{n-1}b_{n-1} + b_{n-2} = 0 \tag{2.30}$$

$$-a_{n-2}b_{n-1} + b_{n-3} = 0 \tag{2.31}$$

$$-a_{n-3}b_{n-1} + b_{n-4} = 0 \tag{2.32}$$

$$\vdots$$

$$-a_1 b_{n-1} + b_0 = 0 \tag{2.33}$$

$$\text{and } -a_0 b_{n-1} = 1 \tag{2.34}$$

From equation (2.34)

$$b_{n-1} = \left(\frac{-1}{a_{n-1}}\right)$$

Substituting $b_{n-1} = \left(\frac{-1}{a_{n-1}}\right)$ in rest of the equations
$b_0 = \left(\frac{-a_1}{a_{n-1}}\right)$, $b_1 = \left(\frac{-a_2}{a_{n-1}}\right) \cdots b_{n-3} = \left(\frac{-a_{n-2}}{a_{n-1}}\right)$ and $b_{n-2} = \left(\frac{-a_{n-1}}{a_{n-1}}\right)$

2.4 Proposed theorem on eigen function of a matrix in GF(p^n)

THEOREM 2.2

If the eigen function of an r × r matrix $A(x)$ having the elements in GF(p^n) is a primitive polynomial, then $A^k(x) = I$, where $k = n^r - 1$.

Proof - The eigen function of $r \times r$ matrix is a polynomial of degree r. Since, the matrix is defined in GF(p^n), the polynomial belongs to the extended field. As per the definition of primitive polynomial, it divides $x^{n^r} - 1$. By Sylvester theorem, the eigen function satisfies the matrix equation $A^{n^r}(x) = 1$

2.5 Proposed algorithm for encryption and decryption

(a) Encryption

Step 1. Generate a self-invertible matrix 'A' by any one of the methods mentioned in section 2.1 and section 2.2.

Step 2. Select 'n' degree polynomials with non-zero roots arbitrarily.

Step 3. Generate a matrix 'B' with eigen factors derived from Step 2 and using the method discussed in section 2.3.

Step 4. Determine the key matrix 'C' for encryption by the relation $C = A.B.A$.

Step 5. Divide the image into 8 × 8 blocks.

Step 6. Encrypt each block by the key matrix.

Step 7. Combine these blocks to form the encrypted image.

(a) Decryption

Step 1. Divide the encrypted image into 8 × 8 blocks.

Step 2. Generate decryption matrix 'D' by the relation $D = A.B^{-1}.A$ as discussed in section 2.3.2.

Step 3. Decrypt each block using the decryption matrix.

Step 4. Form the decrypted image by combining these blocks.

2.6 Cryptanalysis

Number of 2×2 invertible matrices that exist in Z_p is equal to $(p-2)(p-1)^3$. If Hill cipher is used for symmetric key encryption, the intruder has to test $(p-2)(p-1)^3$ number of matrices in order to find the key matrix. When the dimension of the key matrix is increased and p is taken as a large number, the number of matrices that the intruder needs to test increases exponentially. In this method the computation time of finding the inverse of key matrix (by Gauss-Seidel method) is very large. So, it becomes very difficult for the intruder to find the key matrix. But the receiver also needs to find the inverse of the key matrix which will take a considerable amount of time. In order to decrease the computational time of decryption and maintain the robustness of the encryption, a new asymmetric key encryption technique using self-invertible matrix is proposed.

Number of 2×2 self-invertible matrices that exist in Z_p can be analytically found out to be $\frac{(p-2)^2}{2}$. As mentioned earlier, when dimension of the matrix is increased to N and p is taken a large number, the intruder has to test $\frac{(p-2)^N}{2}$ number of matrices, which is very large. The proposed algorithm uses asymmetric key technique for encryption. So, in order to decrypt the image, the intruder has to find the inverse of public key which is very difficult to find as explained previously. But the computation time of decryption reduces considerably by the proposed decryption algorithm.

It is also difficult to predict the key by cipher-text-attack as the image is completely scrambled leaving no trace of the original image. The statistical property of the image is also lost which is demonstrated in the histogram of the encrypted image.

From the above analysis it is evident that one cannot find the key by plain-text/cipher-text attack. Moreover image encryption by block transformation technique increases the difficulty level.

Above cryptanalysis also holds good for encryption in GF(2^8) and GF(13^2).

2.7 Results

The proposed algorithm for encryption and decryption is validated using simulation technique. In the simulation studies, image of 256×256 pixels with 8 bit encoding for each pixel is considered.

2.7.1 Simulation result of encryption in GF(p) ($p = 251$)

Let A be a self-invertible matrix selected randomly in GF(p).

$$A = \begin{bmatrix} 83 & 48 & 155 & 157 & 231 & 91 & 43 & 47 \\ 231 & 138 & 119 & 173 & 49 & 123 & 113 & 2 \\ 136 & 79 & 191 & 226 & 89 & 7 & 105 & 215 \\ 82 & 57 & 197 & 146 & 150 & 191 & 203 & 45 \\ 133 & 44 & 158 & 6 & 153 & 3 & 17 & 83 \\ 112 & 211 & 28 & 70 & 155 & 35 & 227 & 181 \\ 45 & 147 & 204 & 162 & 6 & 81 & 86 & 212 \\ 189 & 6 & 23 & 152 & 45 & 183 & 41 & 170 \end{bmatrix}$$

The private key matrix B (in GF(p)) has been generated by taking eigen factors $x - 31$, $x - 7$, $x - 191$, $x^3 + x^2 + 1$, $x - 47$ and $x - 127$ as defined in section 2.3. So, the private key B can be presented as

$$B = \begin{bmatrix} 31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 191 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 47 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 127 \end{bmatrix}$$

The public key matrix C (in GF(p)) is generated by using the relation $C = A.B.A$

$$C = \begin{bmatrix} 45 & 95 & 198 & 73 & 140 & 150 & 16 & 25 \\ 147 & 22 & 181 & 104 & 147 & 57 & 165 & 240 \\ 29 & 202 & 104 & 205 & 70 & 228 & 153 & 74 \\ 98 & 196 & 173 & 107 & 23 & 97 & 9 & 188 \\ 68 & 150 & 78 & 25 & 213 & 4 & 88 & 12 \\ 201 & 214 & 113 & 70 & 210 & 173 & 81 & 27 \\ 129 & 44 & 100 & 120 & 129 & 189 & 161 & 156 \\ 8 & 63 & 52 & 165 & 12 & 56 & 77 & 79 \end{bmatrix}$$

The inverse of public key D (in GF(p)) is generated by using the relation $D = A.B^{-1}.A$

$$D = \begin{bmatrix} 131 & 224 & 48 & 1 & 225 & 161 & 127 & 133 \\ 106 & 184 & 221 & 181 & 25 & 41 & 145 & 248 \\ 158 & 189 & 81 & 212 & 65 & 197 & 6 & 44 \\ 88 & 195 & 40 & 103 & 146 & 52 & 38 & 234 \\ 185 & 58 & 41 & 174 & 104 & 131 & 187 & 157 \\ 4 & 132 & 177 & 54 & 138 & 51 & 136 & 10 \\ 213 & 245 & 225 & 122 & 180 & 149 & 43 & 173 \\ 221 & 146 & 173 & 191 & 226 & 131 & 133 & 120 \end{bmatrix}$$

256×256 cameraman image and lena image shown in Figure 2.1 are considered to carry out the simulation. Their histograms are presented in Figures 2.2 and 2.3 respectively.

Figure 2.1: Original cameraman and lena image used for encryption

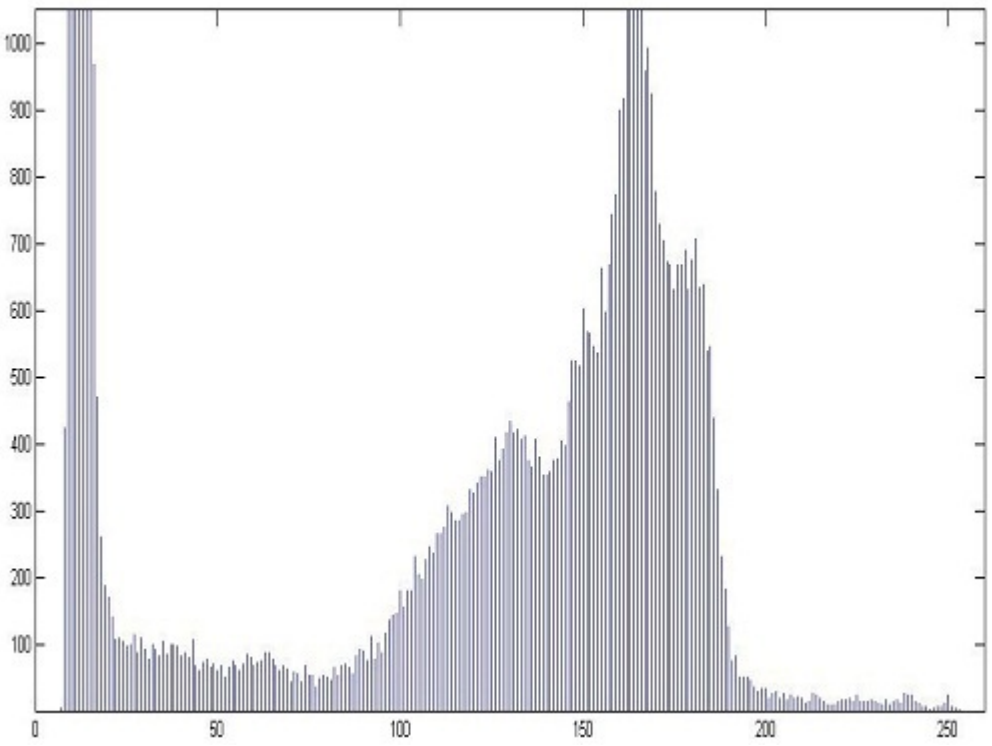

Figure 2.2: Histogram of original cameraman image

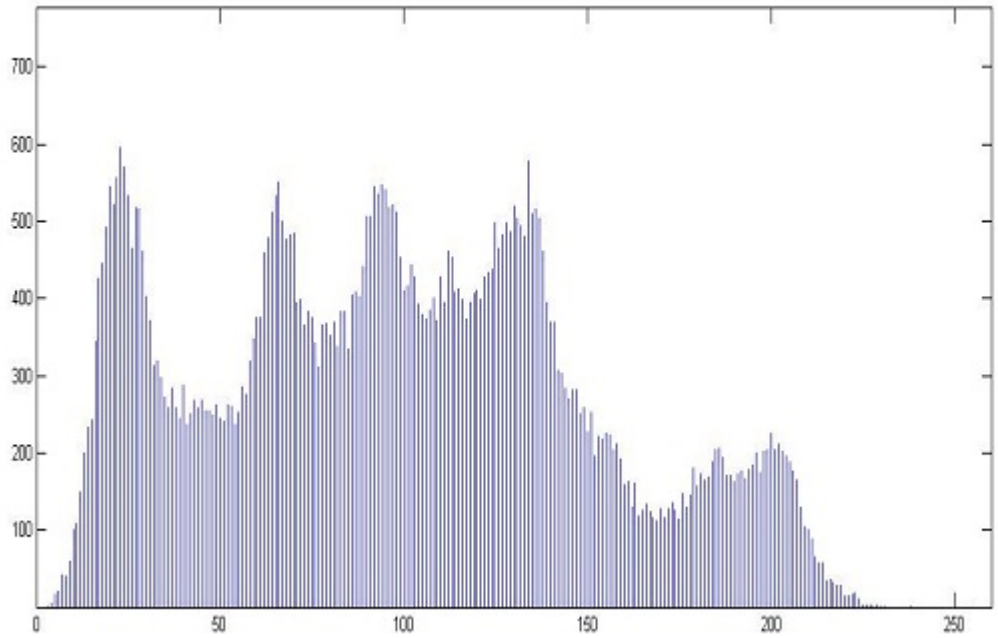

Figure 2.3: Histogram of original lena image

Since encryption is done in $GF(\mathrm{p})$ with $p = 251$, pixels of original image beyond 250 are rounded to 250. The images rounded to 250 pixels and their corresponding histograms are shown in Figures 2.4, 2.5 and 2.6 respectively.

Figure 2.4: Cameraman and lena image with pixels ≤ 250

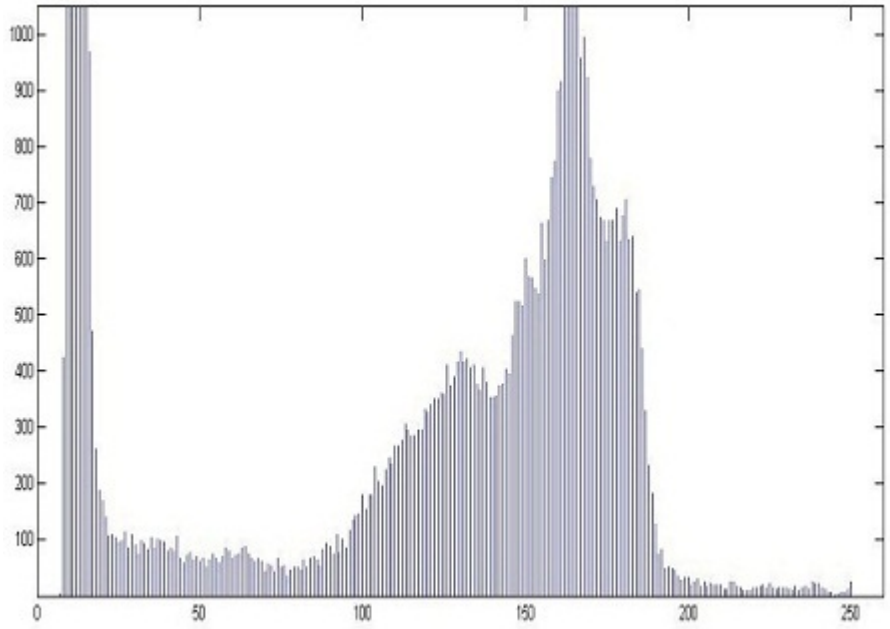

Figure 2.5: Histogram of cameraman image with pixels ≤ 250

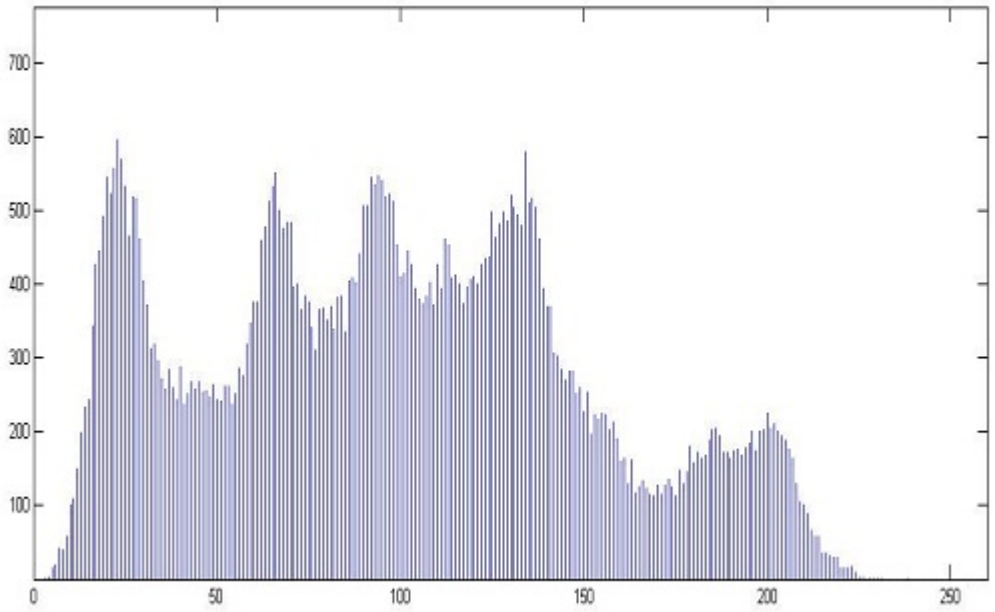

Figure 2.6: Histogram of lena image with pixels ≤ 250

The encryption algorithm proposed in section 2.5 is used to encrypt the cameraman and lena images which are rounded to 250 pixels. After encryption, the images of cameraman and lena and their corresponding histograms are shown in Figures 2.7, 2.8 and 2.9 respectively. On comparing the encrypted images with the original images, it can be seen that the encrypted images and their histograms

do not bear any resemblance with the original images and their histograms.

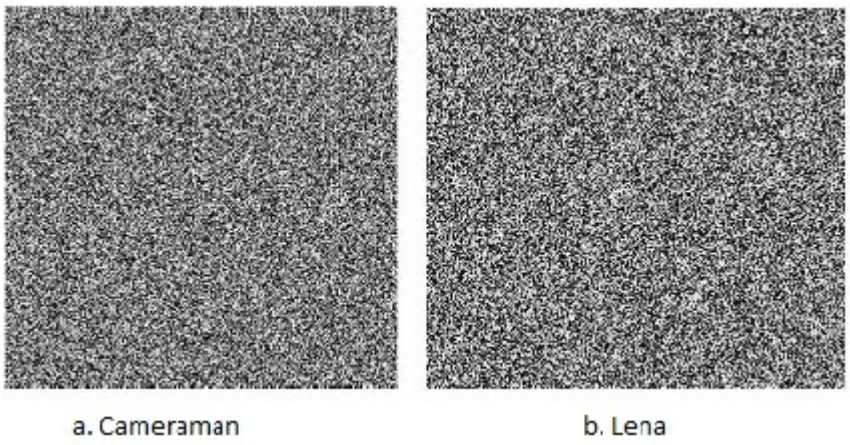

Figure 2.7: Encrypted image of cameraman and lena in GF(p) (where $p = 251$) using encryption algorithm in section 2.5

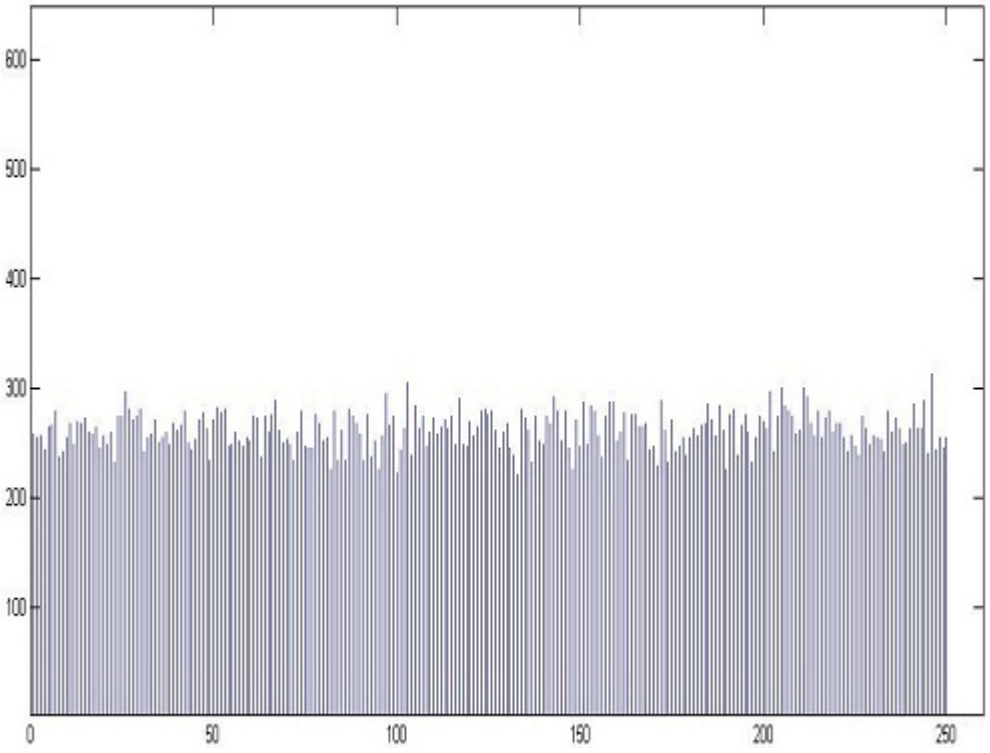

Figure 2.8: Histogram of encrypted image of cameraman in GF(p) (where $p = 251$) using encryption algorithm in section 2.5

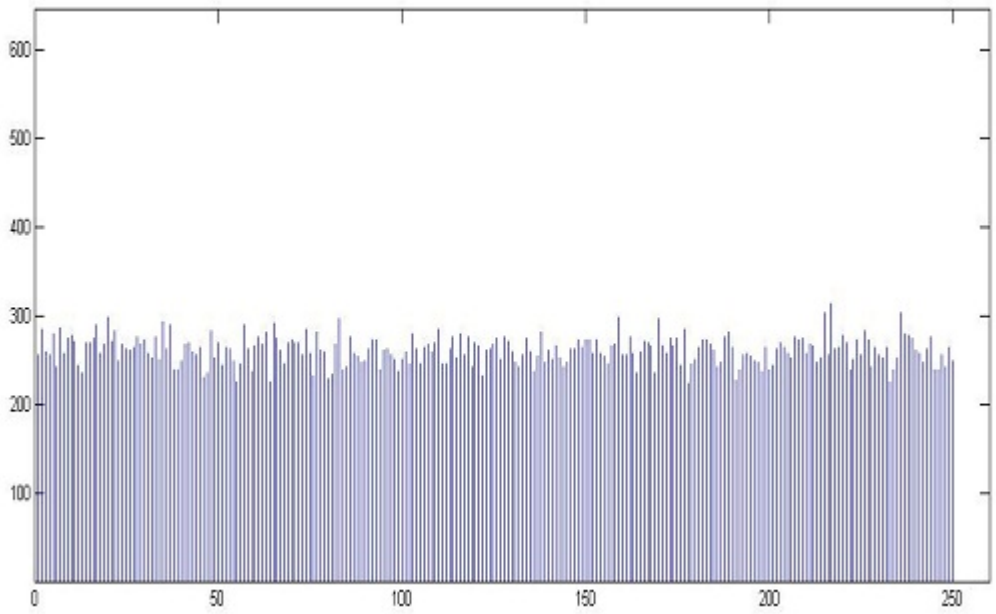

Figure 2.9: Histogram of encrypted image of lena in GF(p)
(where $p = 251$) using encryption algorithm in section 2.5

The encrypted images are decrypted by using the decryption algorithm presented in section 2.5. The decrypted images and their corresponding histograms are shown in Figures 2.10, 2.11 and 2.12 respectively which are same as the original images(rounded to 250 pixels) and their corresponding histograms.

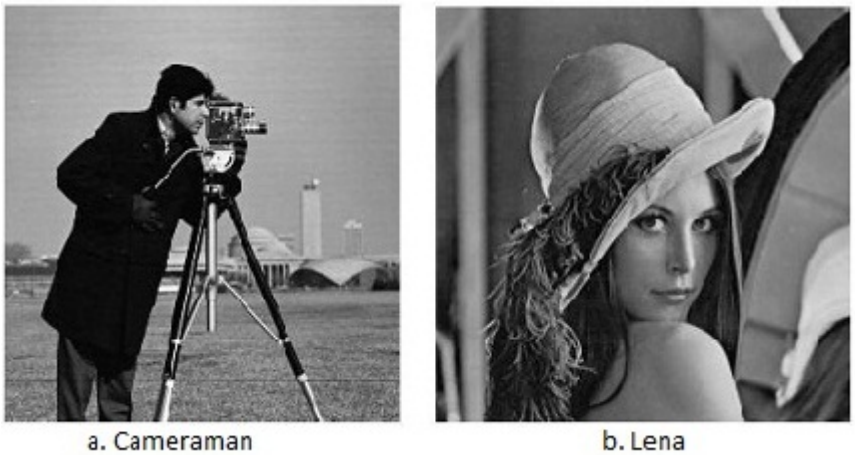

Figure 2.10: Decrypted image of cameraman and lena image in GF(p)
(where $p = 251$) using decryption algorithm in section 2.5

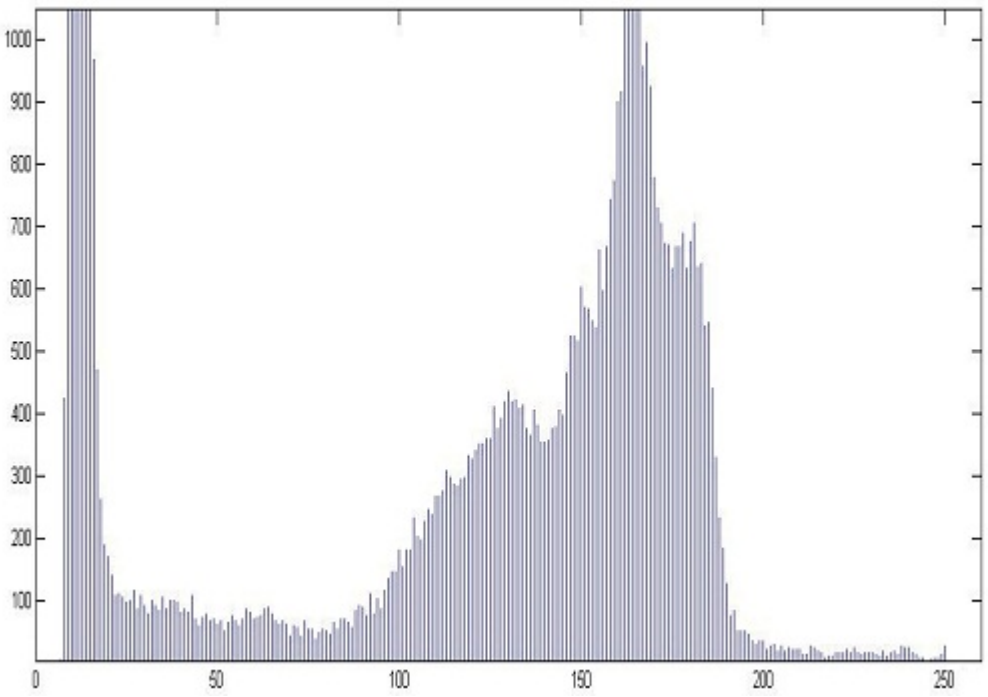

Figure 2.11: Histogram of decrypted image of cameraman in GF(p) (where $p = 251$) using decryption algorithm in section 2.5

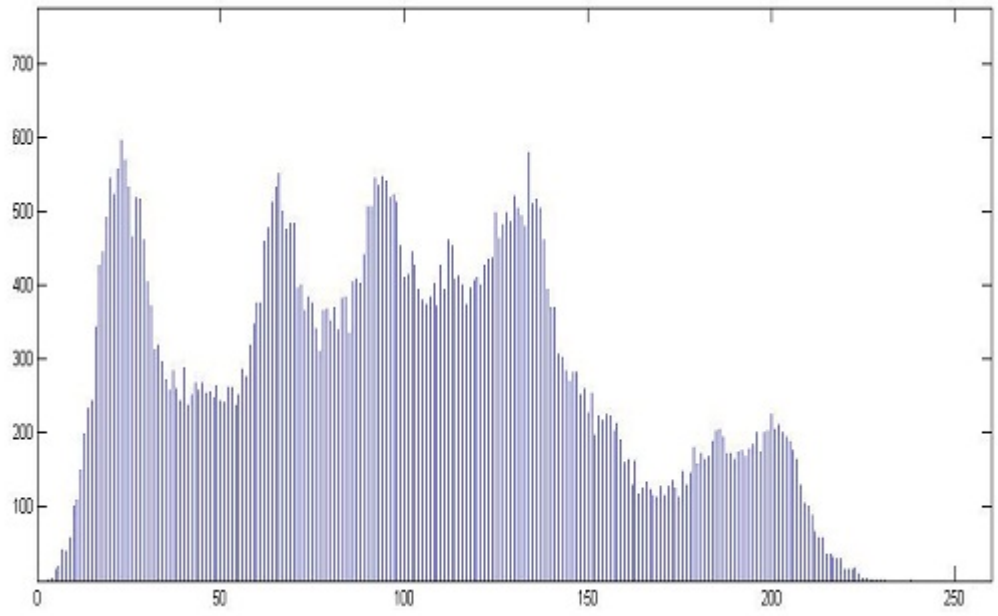

Figure 2.12: Histogram of decrypted image of lena in GF(p) (where $p = 251$) using decryption algorithm in section 2.5

2.7.2 Simulation result of encryption in $GF(2^8)$

The primitive polynomial for 8-bit data encryption in GF(2^8) is $D^8 + D^4 + D^3 + D^2 + 1 = 285$.

Let A be a self-invertible matrix selected randomly in GF(2^8) whose decimal equivalent is given here.

$$A = \begin{bmatrix} 126 & 44 & 206 & 129 & 243 & 125 & 136 & 167 \\ 6 & 171 & 226 & 67 & 181 & 12 & 32 & 52 \\ 84 & 182 & 159 & 42 & 244 & 72 & 112 & 132 \\ 217 & 227 & 104 & 172 & 75 & 172 & 192 & 232 \\ 145 & 187 & 60 & 100 & 137 & 229 & 231 & 72 \\ 244 & 180 & 162 & 231 & 62 & 165 & 53 & 94 \\ 64 & 47 & 28 & 148 & 133 & 234 & 4 & 1 \\ 20 & 154 & 139 & 181 & 173 & 55 & 238 & 206 \end{bmatrix}$$

The private key matrix B (in $GF(2^8)$) has been generated by taking eigen factors $x - (D + 1)$, $x - (D^2 + D + 1)$, $x - (D^3 + D^2 + 1)$, $x - (D^4 + D^3 + 1)$, $x - (D^4 + D^3 + D^2 + D + 1)$ and $x^3 - x^2 - x - D^3$ as defined in Section 2.5. So, decimal equivalent of the private key B can be presented as

$$B = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 31 \end{bmatrix}$$

The public key matrix C (in $GF(2^8)$) is generated by using the relation $C = A.B.A$. The decimal equivalent of C is given as below.

$$C = \begin{bmatrix} 100 & 65 & 30 & 107 & 19 & 171 & 106 & 150 \\ 28 & 230 & 67 & 19 & 79 & 204 & 218 & 30 \\ 114 & 10 & 231 & 226 & 210 & 72 & 36 & 112 \\ 104 & 91 & 169 & 104 & 72 & 187 & 61 & 111 \\ 208 & 232 & 83 & 103 & 244 & 125 & 249 & 181 \\ 109 & 128 & 89 & 106 & 255 & 241 & 161 & 212 \\ 110 & 139 & 119 & 245 & 120 & 169 & 46 & 251 \\ 101 & 229 & 186 & 144 & 160 & 217 & 53 & 40 \end{bmatrix}$$

The inverse of public key D (in $GF(2^8)$) is generated by using the relation $D = A.B^{-1}.A$. The decimal equivalent of D is given as below.

$$D = \begin{bmatrix} 191 & 95 & 113 & 197 & 203 & 54 & 130 & 120 \\ 13 & 133 & 48 & 68 & 175 & 18 & 68 & 67 \\ 57 & 39 & 191 & 219 & 236 & 45 & 92 & 5 \\ 144 & 174 & 66 & 229 & 250 & 128 & 27 & 230 \\ 53 & 239 & 41 & 246 & 106 & 56 & 20 & 69 \\ 73 & 159 & 67 & 2 & 46 & 144 & 145 & 198 \\ 252 & 154 & 195 & 233 & 107 & 167 & 146 & 140 \\ 57 & 116 & 197 & 42 & 235 & 13 & 13 & 181 \end{bmatrix}$$

The encryption algorithm proposed in section 2.5 is used to encrypt the cameraman and lena images in $GF(2^8)$. The encrypted images and their corresponding histograms are shown in Figures 2.13, 2.14 and 2.15 respectively. On comparing the encrypted images with the original images, it can be seen that the encrypted images and their corresponding histograms do not bear any resemblance with the original images and their histograms.

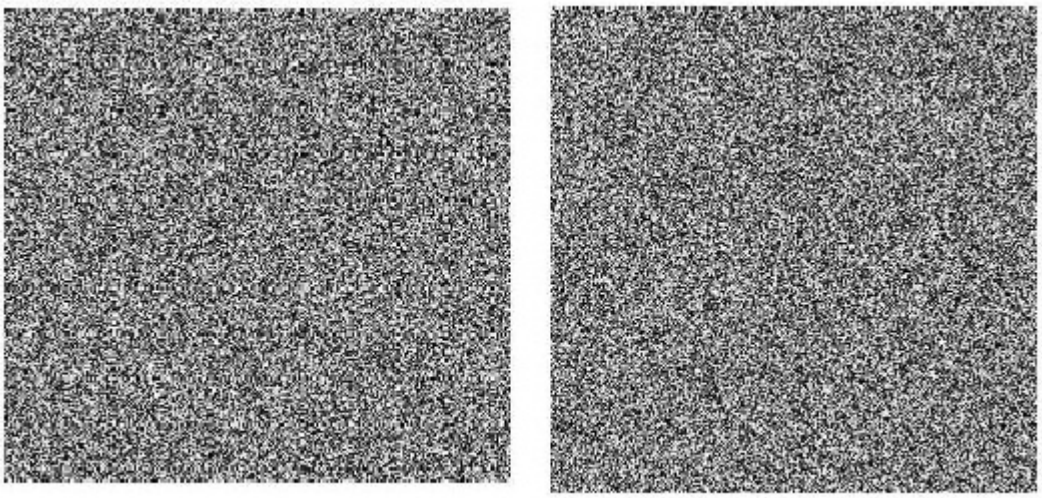

Figure 2.13: Encrypted images of cameraman and lena in $GF(2^8)$ using encryption algorithm in section 2.5

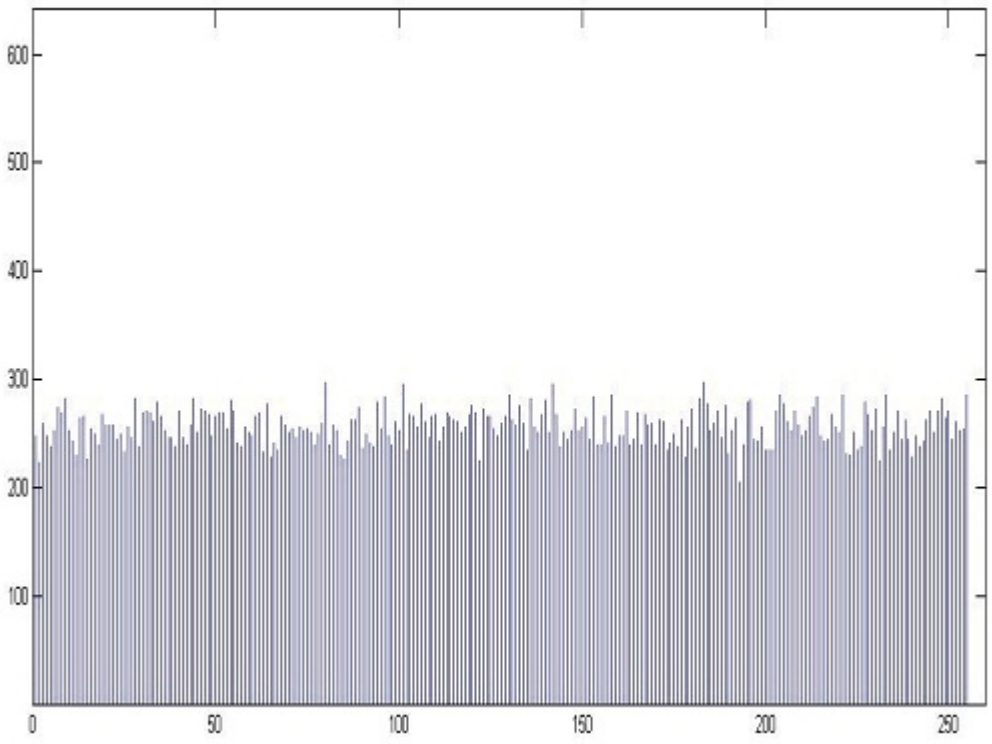

Figure 2.14: Histogram of encrypted image of cameraman in $GF(2^8)$ using encryption algorithm in section 2.5

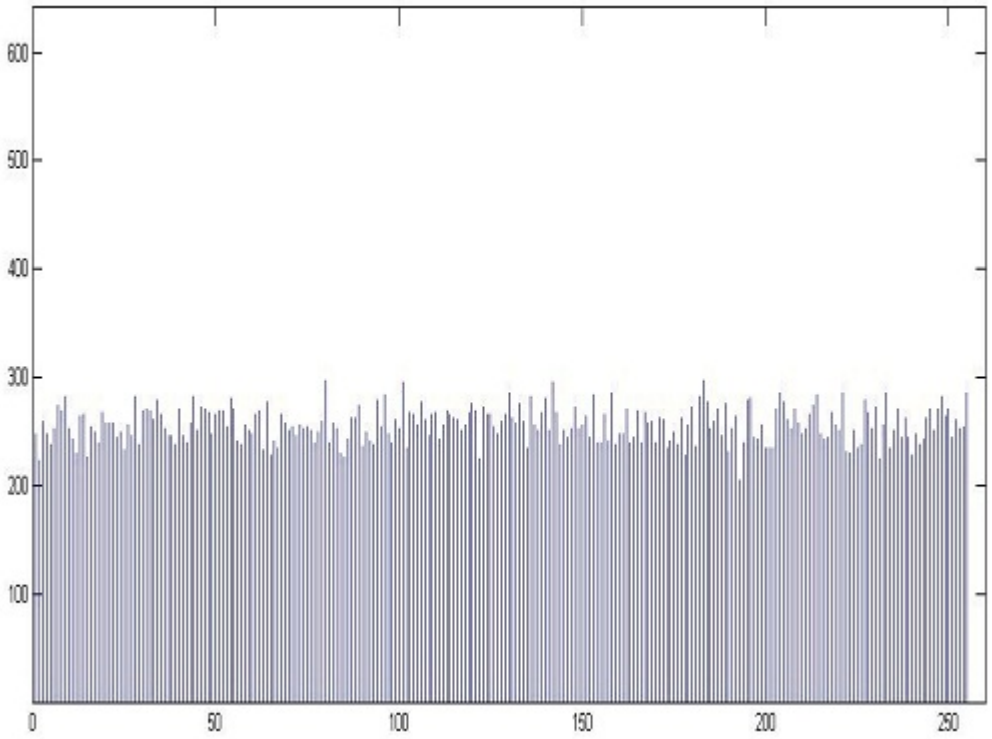

Figure 2.15: Histogram of encrypted image of lena in $GF(2^8)$ using encryption algorithm in section 2.5

The encrypted images are decrypted by using the decryption algorithm presented in section 2.5. The decrypted images and their histograms are shown in Figures 2.16, 2.17 and 2.18 respectively which are same as the original images and their histograms.

Figure 2.16: Decrypted image of cameraman and lena in $GF(2^8)$ using decryption algorithm in section 2.5

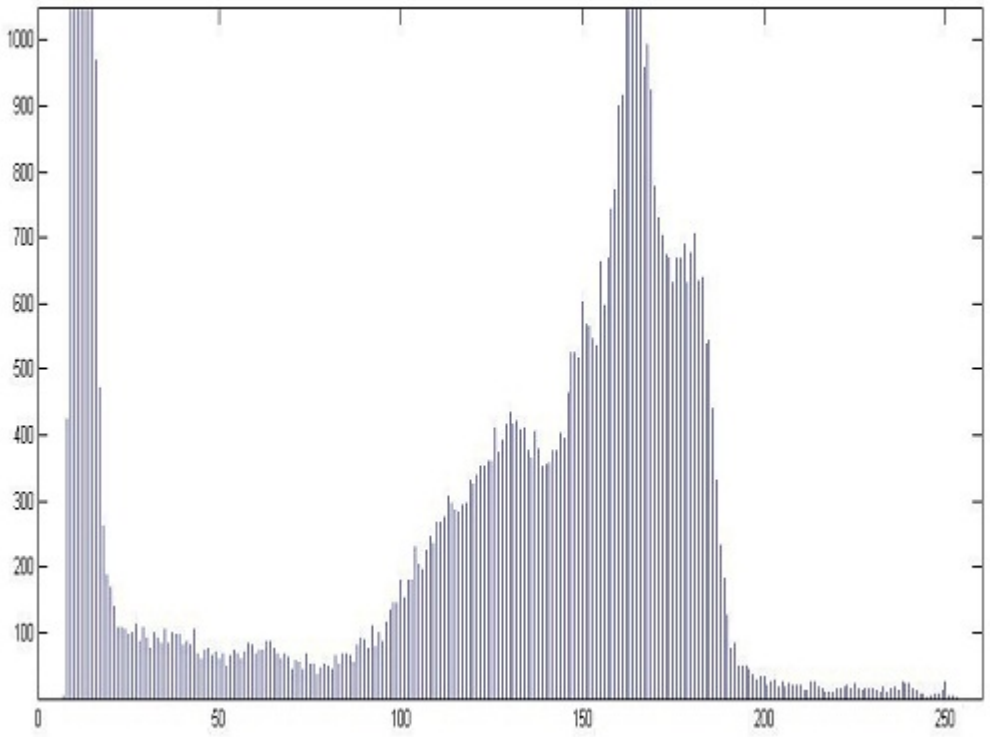

Figure 2.17: Histogram of decrypted image of cameraman in $GF(2^8)$ using decryption algorithm in section 2.5

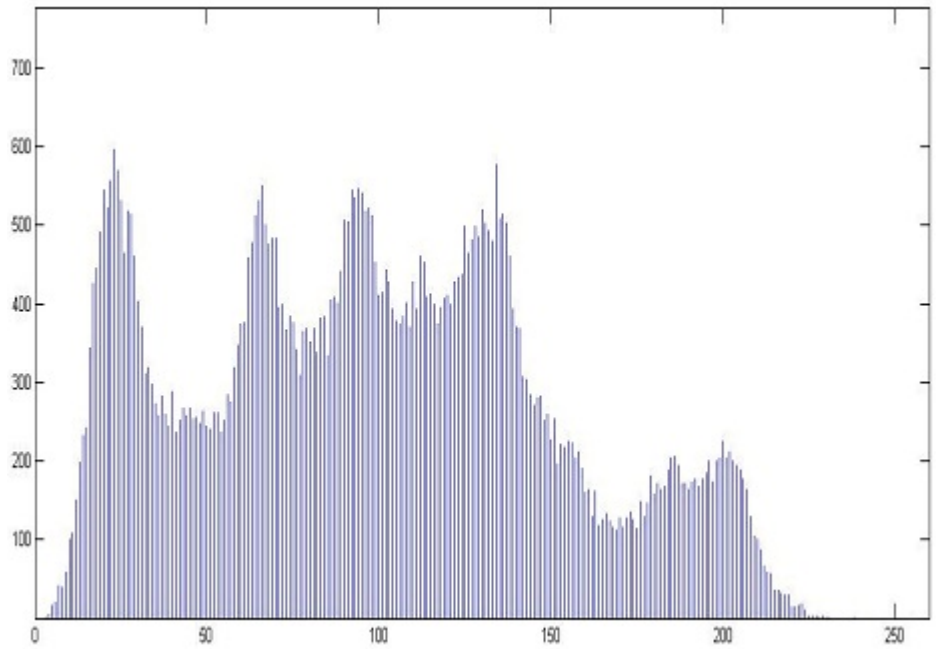

Figure 2.18: Histogram of decrypted image of lena in $GF(2^8)$ using decryption algorithm in section 2.5

2.7.3 Simulation result of encryption over GF(p^n)

Encryption of cameraman and lena images in GF(2^8) is presented in previous section. In this section, encryption in GF(p^n) is presented considering $p = 13$ and $n = 2$, the primitive polynomial over GF(p^n) $D^2 + D^1 + 2 = 184$ has been considered for encryption. Let A be a self-invertible matrix selected randomly in GF(p^n) whose decimal equivalent is given here.

$$A = \begin{bmatrix} 24 & 97 & 79 & 1 & 15 & 40 & 14 & 144 \\ 76 & 159 & 71 & 13 & 11 & 157 & 40 & 65 \\ 74 & 78 & 143 & 91 & 117 & 19 & 0 & 168 \\ 0 & 162 & 129 & 16 & 110 & 37 & 94 & 8 \\ 24 & 110 & 134 & 130 & 70 & 70 & 8 & 118 \\ 108 & 81 & 166 & 90 & 25 & 80 & 26 & 68 \\ 0 & 92 & 14 & 151 & 89 & 76 & 142 & 73 \\ 0 & 15 & 107 & 70 & 140 & 114 & 64 & 83 \end{bmatrix}$$

The private key matrix B (in $GF(p^n)$) has been generated by taking eigen factors $x-1$, $x-(10D+1)$, $x-(9D+1)$, x^3-x^2-x-8, $x-(4D+1)$ and $x-(6D+1)$ as defined in Section 2.5. So, decimal equivalent of the private key B can be presented as

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 131 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 118 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 53 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 79 \end{bmatrix}$$

The public key matrix C (in $GF(p^n)$) is generated by using the relation

$C = A.B.A$. The decimal equivalent of C is given as below.

$$C = \begin{bmatrix} 94 & 121 & 160 & 120 & 4 & 110 & 25 & 156 \\ 17 & 108 & 42 & 121 & 55 & 33 & 94 & 59 \\ 147 & 36 & 34 & 79 & 36 & 3 & 89 & 150 \\ 88 & 164 & 68 & 33 & 21 & 69 & 23 & 89 \\ 155 & 31 & 119 & 29 & 160 & 126 & 135 & 71 \\ 160 & 72 & 147 & 155 & 30 & 149 & 161 & 165 \\ 152 & 52 & 122 & 100 & 59 & 94 & 59 & 110 \\ 148 & 17 & 74 & 81 & 152 & 81 & 113 & 123 \end{bmatrix}$$

The inverse of public key D (in $GF(p^n)$) is generated by using the relation $D = A.B^{-1}.A$. The decimal equivalent of D is given as below.

$$D = \begin{bmatrix} 25 & 112 & 102 & 13 & 44 & 97 & 94 & 37 \\ 60 & 109 & 44 & 96 & 71 & 62 & 165 & 19 \\ 162 & 21 & 127 & 82 & 57 & 120 & 2 & 87 \\ 154 & 46 & 103 & 40 & 59 & 22 & 114 & 137 \\ 157 & 82 & 65 & 130 & 40 & 43 & 17 & 70 \\ 96 & 81 & 139 & 162 & 104 & 117 & 55 & 105 \\ 168 & 104 & 131 & 3 & 130 & 108 & 57 & 90 \\ 107 & 112 & 167 & 111 & 4 & 1 & 20 & 124 \end{bmatrix}$$

Since encryption is done in GF(p^n) with $p = 13$ and $n = 2$, pixels of original image beyond 168 are rounded to 168. The image rounded to 168 pixels and its histogram are shown in Figures 2.19, 2.20 and 2.21 respectively.

a. Cameraman b. Lena

Figure 2.19: Cameraman and lena images with pixels ≤ 168

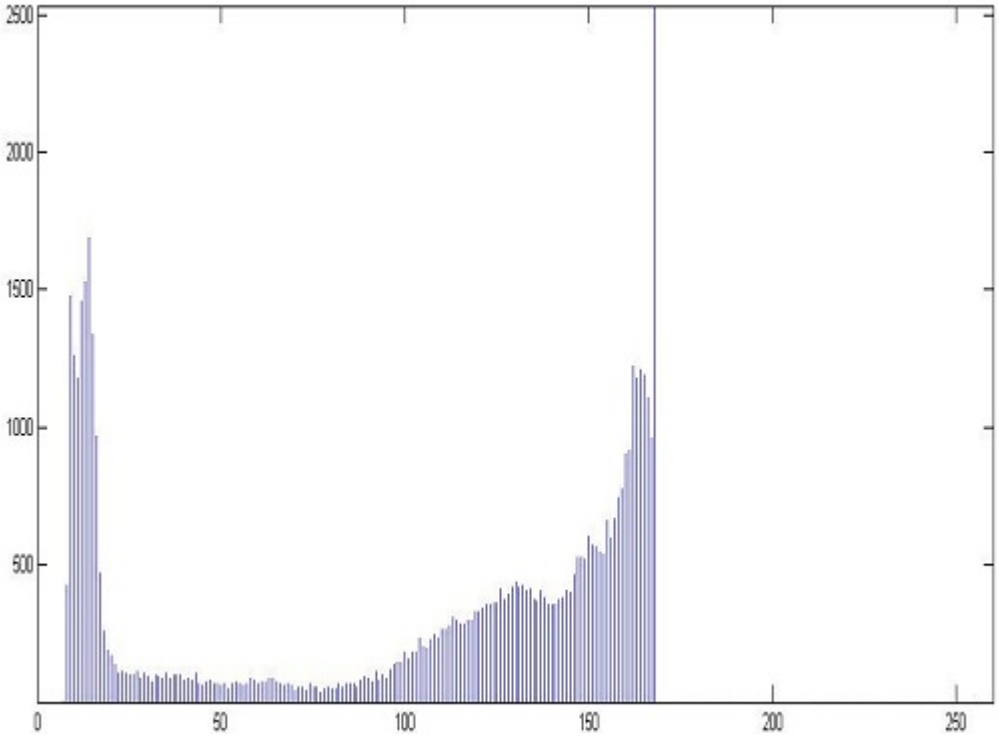

Figure 2.20: Histogram of cameraman image with pixels ≤ 168

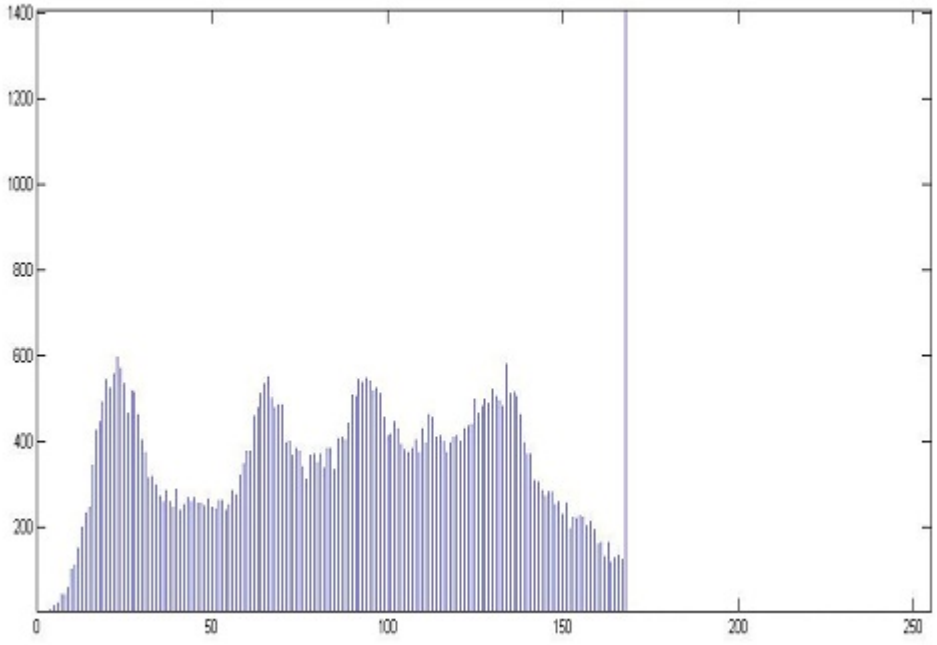

Figure 2.21: Histogram of lena image with pixels ≤ 168

The encryption algorithm proposed in section 2.5 is used to encrypt the cam-

eraman and lena images which are rounded to 168 pixels. The encrypted images and their corresponding histograms are shown in Figures 2.22, 2.23 and 2.24 respectively. On comparing the encrypted images with the original images, it can be seen that the encrypted images and their histograms do not bear any resemblance with the original images and their histograms.

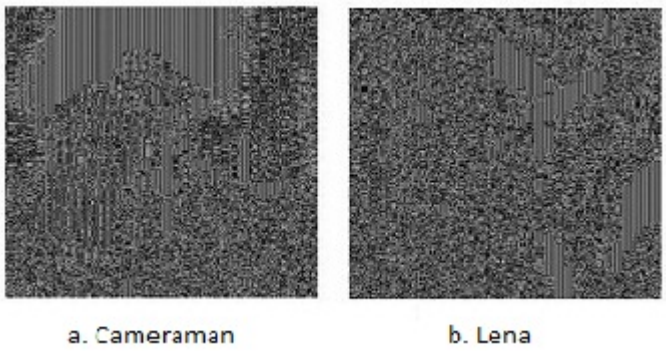

Figure 2.22: Encrypted image of cameraman and lena in GF(13^2) using encryption algorithm in section 2.5

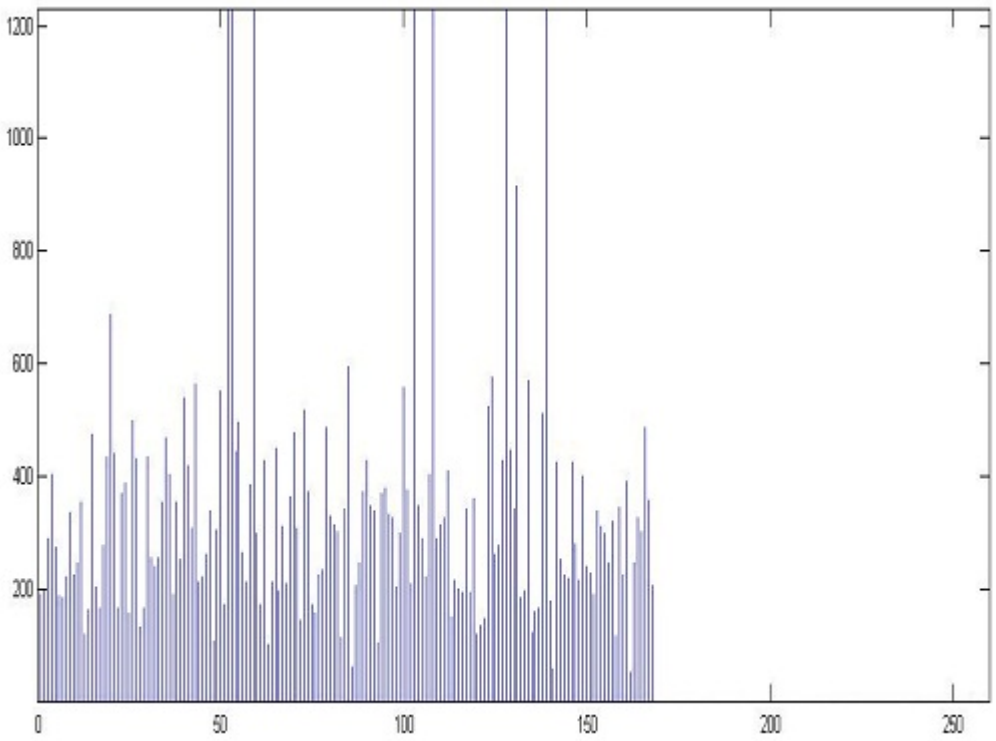

Figure 2.23: Histogram of encrypted image of cameraman in GF(13^2) using encryption algorithm in section 2.5

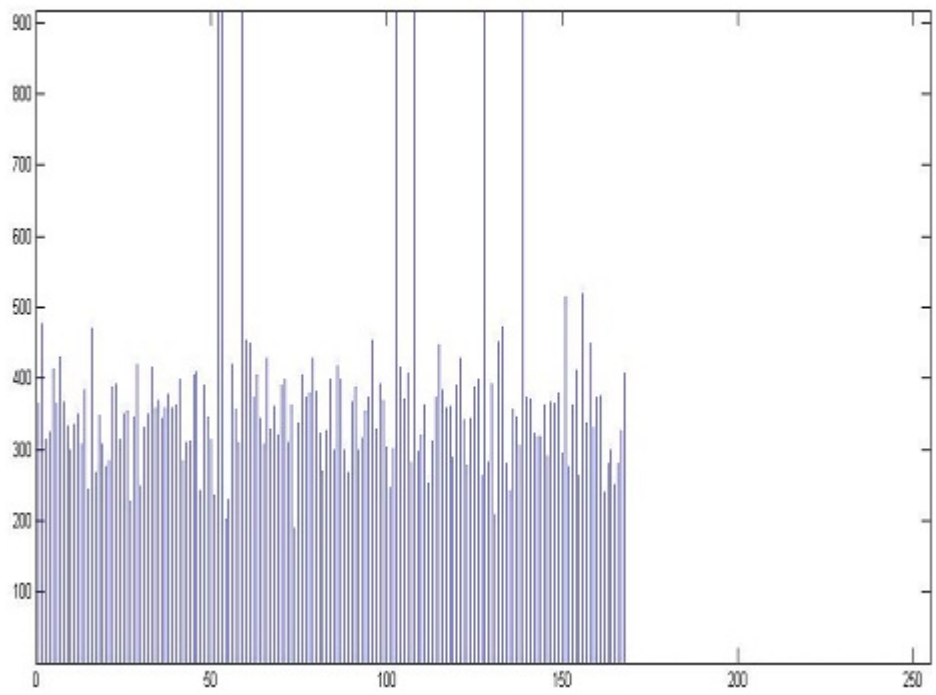

Figure 2.24: Histogram of encrypted image of lena in GF(13^2) using encryption algorithm in section 2.5

The encrypted image is decrypted by using the decryption algorithm presented in section 2.5. The decrypted images and their histograms are shown in Figures 2.25, 2.26 and 2.27 respectively which are same as the original images (rounded to 250 pixels) and their histograms.

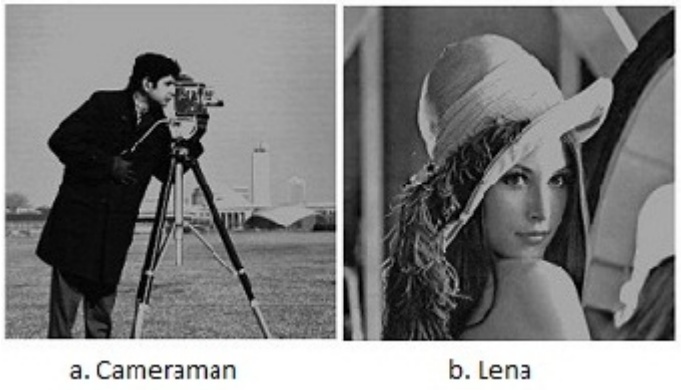

Figure 2.25: Decrypted image of cameraman and lena in GF(13^2) using decryption algorithm in section 2.5

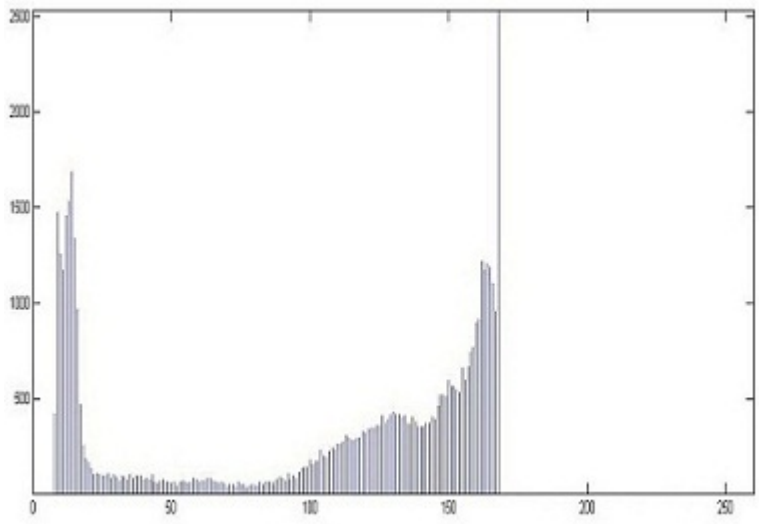

Figure 2.26: Histogram of decrypted image of cameraman GF(13^2) using decryption algorithm in section 2.5

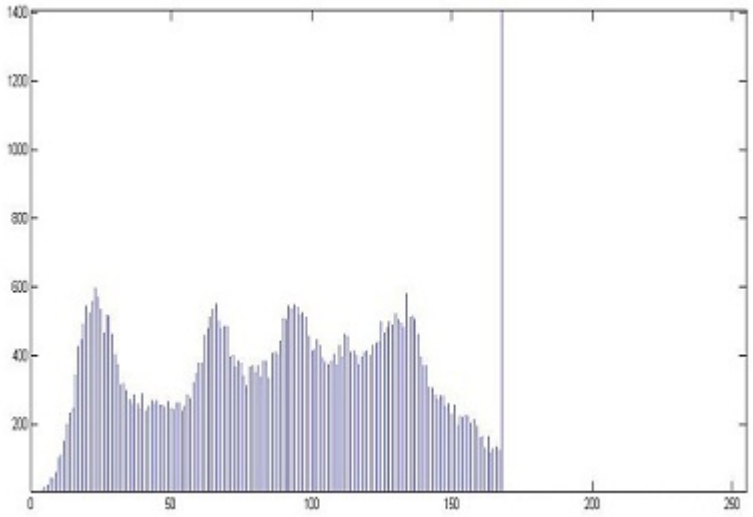

Figure 2.27: Histogram of decrypted image of lena in GF(13^2) using decryption algorithm in section 2.5

The proposed algorithm is also tested with other images. Some of the simulation results are shown in Figure 2.28, where Figure 2.28(a,b,c) represent the original images of Baboon, Barbara and Crowd. The corresponding histograms of the original images are shown in Figure 2.28(d,e,f). The encryption algorithm proposed in section 2.5 is used for encrypting the images. The encrypted images are shown in Figure 2.28(g,h,i) and their histograms are represented in Figure 2.28(j,k,l).

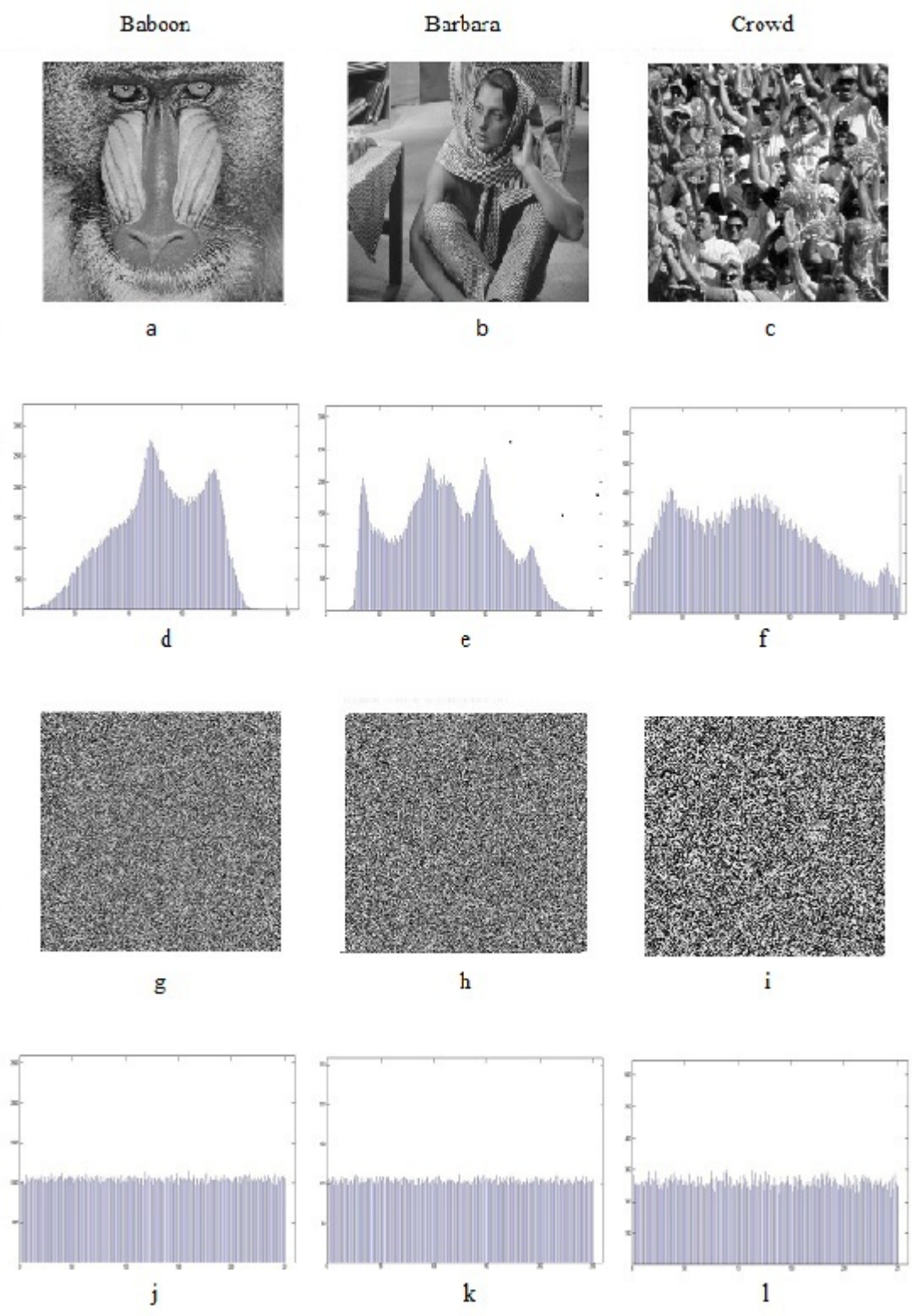

Figure 2.28: Additional simulation results

2.8 Summary

Several methods of generating self-invertible matrices in GF(p), GF(2^8) and GF(p^n) are proposed in this chapter. In order to enhance the security, the public key for encryption is taken as the product of self-invertible matrix and another matrix. If the size of the matrix is increased, then decryption will be difficult. It has been shown through theorems that eigen factors are very much essential to characterize matrices in Galois field. Specifically, these principles can be used to determine inverse of matrices by repeated multiplication of the same matrix. Results have been validated through encryption of images using simulation techniques.

Chapter 3

Discrete Orthogonal Transform and Image Encryption

Chapter 3

Discrete Orthogonal Transform and Image Encryption

Transforms play major role in the field of engineering and applied science. It is easier to work on the transformed data and lots of useful information can be obtained from them. Fourier Transform, Cosine Transform, Sine Transform, etc are the common transforms based on sinusoids because of the cyclic nature of the signals. Walsh Transform and Walsh Hadamard Transform are simpler transforms as compared to sinusoid-based transforms, in which the kernel is made of $+1$ and -1. These transforms can be implemented without any additional multiplication operation. Hence, Walsh and Walsh Hadamard transforms are much faster than Fast Fourier Transform. Modular Orthogonal Transforms such as Walsh Transform, Walsh Hadamard Transform, Discrete Cosine Transform, Discrete Sine Transform, Discrete Fourier Transform have been used for encryption of image [28–31].

In this chapter, orthogonal matrices with elements belonging to finite field i.e. $GF(p)$, $GF(2^n)$ and $GF(p^n)$ have been introduced and several methods of generating such matrices have been suggested. These orthogonal matrices are

used as key matrices for encryption. To make encryption more secure another key matrix is used which is generated by method similar to section 2.3.

3.1 Basic Theory

Let the image E be represented as a $M \times N$ matrix of integer numbers. An image transform can generally process either the whole image or some part of the image [28–31]. Transform matrices P and Q of dimension $M \times M$ and $N \times N$, respectively, are used to transform E into a matrix T of dimension $M \times N$.

$$T = PEQ \tag{3.1}$$

If P and Q are non-singular, P^{-1} and Q^{-1} exist and the inverse transform can be computed as

$$E = P^{-1}TQ^{-1} \tag{3.2}$$

Few terms and formulae are required for better understanding of this theory. Let M^T represent the transpose of the matrix M.

• M is symmetric if $M = M^T$

• M is orthogonal if $M^T M = kI$, where I is the identity matrix and k is any number. If $k = 1$, then M is orthonormal.

• For any real and orthogonal matrix M, $\mathrm{M}^{-1} = \frac{1}{k}M^T$.

3.1.1 Hadamard Transform

Hadamard matrices are square matrices with entries +1 or -1 and have orthogonal row vectors and orthogonal column vectors [33,54]. The Hadamard Transform was first described by Hadamard in 1893. It grew out of matrix theory and has been used widely in image processing, image compression and pattern recognition.

Hadamard matrix operator is used to transform an image which is similar to the two-dimensional Fourier transform [7].

If [f] represents the image and [F] the transformed image, the Hadamard Transform is represented as $[F] = H_N[f]H_N/N$, where H_N represents a $N \times N$ Hadamard matrix with element values either - 1 or + 1. The inverse Hadamard Transform is given by $[f] = H_N[F]H_N/N$

For N=2, the Hadamard matrix is defined as

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Then the Hadamard matrix of order 2N is generated in terms of the Hadamard matrix of order N using the Kronecker product [7], as $H_{2N} = H_2 \times H_N$

For N=2, H_4 matrix is obtained as

$$H_{2\times 2} = H_2 \otimes H_2 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Similarly $H_8, H_{16}, H_{32}, H_{64}, H_{128}, H_{256}$ matrices are obtained.

Properties

• Hadamard kernel is symmetric and orthogonal over finite field.

• Sequence of a row or a column of the kernel is given by the decimal representation of the Gray code of the bit reversed binary values of the corresponding row or the column index.

• The determinant of the minimum kernel is 2.

• The determinant of any other higher kernel of size $2^n \times 2^n \; is \; 2^{n2^{n-1}}$ for $n > 1$.

3.1.2 Walsh Transform

Walsh functions are often used in engineering applications, including communication systems and digital image processing. Walsh Transform was developed by Walsh in 1923 by modification of Hadamard Transform and sometimes it is called Walsh-Hadamard transform. M. A Karagodin et.al used the following Hadamard Transform for image compression [13].

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

The characteristics of the columns of H_4 are the sign transitions (a transition is defined as a -1 to 1 or 1 to -1 change). For H_4 the first column has 0 sign transitions, the second column 3 sign transitions, the third column 1 sign transition, and the fourth column 2 sign transitions. The number of sign changes is referred to as the sequency. If the columns of H_4 are arranged in order of increasing sequence, a Walsh transform matrix w_4 is obtained as [28–31, 55]

$$w_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

Hence, the Walsh Transform matrix is sequency ordered Hadamard transform matrix. The Walsh transform is given by $[F] = W_N[f]W_N/N$

The inverse Walsh transform is computed as $[f] = W_N[F]W_N/N$ similar to the Hadamard Transform matrix. The rows (and columns) of the Walsh Transform matrix are orthogonal and $W_{N-1} = W_N/N$. The L $\times$ L upper left partition of

the transformed image [F] using the Walsh transform corresponds to the lower L sequency components.

Similarly, $W_8, W_{16}, W_{32}, W_{64}, W_{128}, W_{256}$ matrices are obtained.

Properties

• The Walsh transformation kernel is symmetric and orthogonal.

• Each row index or column index gives the sequency of the corresponding row or the column in the kernel.

• The determinant of the minimum kernel is 2.

• And the determinant of any other higher kernel of size $2^n \times 2^n$ is $2^{n.2n-1}$ for $n \geq 2$

Relationship between Walsh-ordered and Hadamard-ordered Transforms

Walsh Ordered	Binary	Reverse Ordered	Gray Ordered (Hadamard ordered)	Decimal
0	000	000	000	0
1	001	100	111	7
2	010	010	011	3
3	011	110	100	4
4	100	001	001	1
5	101	101	110	6
6	110	011	010	2
7	111	111	101	5

3.1.3 The Discrete Cosine Transform

The N × N cosine transform matrix C = c(k, n), also called the Discrete Cosine Transform (DCT) is defined as [20, 21]

$$c(k,n) = \begin{cases} \frac{1}{\sqrt{N}} & \text{k=0 and n} \in \text{[0,N-1]}, \\ \left(\sqrt{\frac{2}{N}}\right) cos\left(\frac{\pi(2n+1)(k)}{2N}\right) & \text{k} \in \text{[1,N-1] and n} \in \text{[0,N-1]}. \end{cases}$$

The one-dimensional DCT of a sequence u(n), $0 \leq n \leq N-1$ is defined as

$$v(k) = \alpha(k) \sum_{n=0}^{N-1} u(n) cos\left(\frac{\pi(2n+1)k}{2N}\right), \ k \in [0, N-1] \tag{3.3}$$

$$\text{where } \alpha(0) = \sqrt{\frac{1}{N}}, \ \alpha(k) = \sqrt{\frac{2}{N}}, k \in [1, N-1] \tag{3.4}$$

The inverse transformation is given by

$$u(n) = \sum_{k=0}^{N-1} \alpha(k) v(k) cos\left(\frac{\pi(2n+1)k}{2N}\right), \ n \in [0, N-1] \tag{3.5}$$

The basis vector of the 8×8 DCT shows the cosine transform of the image scan line. Most of the transform coefficients are small, i.e. maximum energy of the data is packed in a few transform coefficients. The two dimensional cosine transform pair is obtained by substituting A=A∗ =C. The basis images of the 8 × 8 two dimensional cosine transform are the cosine transform of different images.

Properties

• The cosine transform is real and orthogonal, i.e. $C = C^T \Rightarrow C^{-1} = C^T$

• The cosine transform is not real part of the unitary DFT. This can be seen by inspection of C and the DFT matrix F. However, the cosine transform of a sequence is related to the DFT of its symmetric extension.

• The cosine transform is a fast transform. The cosine transform of a vector of N elements can be calculated in O(N log_2N) operations via an N-point, where O is the order of operation.

• The cosine transform has excellent energy compaction for highly correlated data.

3.1.4 The Discrete Sine Transform

The N × N sine transform matrix also called as Discrete Sine Transform (DST) $\Psi = \Psi(k, n)$ is defined as

$$\Psi(k,n) = \frac{1}{N} sin\left(\frac{\pi(k+1)(n+1)}{N+1}\right), \ k \in [0, N-1] \tag{3.6}$$

The sine transform pair of one-dimensional sequences is defined as

$$v(k) = \left(\sqrt{\frac{2}{N+1}}\right) \sum_{n=0}^{N-1} u(n) sin\left(\frac{\pi(k+1)(n+1)}{N+1}\right), \ k \in [0, N-1] \tag{3.7}$$

$$u(n) = \left(\sqrt{\frac{2}{N+1}}\right) \sum_{k=0}^{N-1} v(k) sin\left(\frac{\pi(k+1)(n+1)}{N+1}\right), \ n \in [0, N-1] \tag{3.8}$$

Two dimensional sine transform pair for N × N images is obtained by substituting $A = A^* = A^T = \Psi$ in the basis vectors and the basis images of the sine transform

Properties

• The sine transform is real, symmetric and orthogonal, i.e. $\Psi^* = \Psi = \Psi^T = \Psi - 1$. Thus, the forward and inverse sine transforms are identical.

• The sine transform is not the imaginary part of the unitary DFT. The sine transform of a sequence is related to the DFT of its anti symmetric extension.

• The sine transform is a fast transform. The sine transform (or its inverse) of a vector of N elements can be calculated in $O(Nlog_2N)$ operations via a 2(N+1) point FFT. Typically this requires N+1 = $2p$, i.e. the fast sine transform is usually defined for N=3, 7, 15, 31, 63, 255. Fast sine transform algorithms that do not require complex arithmetic are also possible. In fact, these algorithms are somewhat faster than the FFT and the fast cosine transform algorithms.

The sine transform is close to the KL transform algorithm for Markov sequences, whose boundary values are given. This makes it useful in many image processing problems.

3.2 Proposed methods of generating orthonormal matrices in GF(p)

Several methods of generating orthonormal matrices in GF(p) are proposed in the following sections.

3.2.1 Method I

Let A_2 be a (2×2) orthonormal matrix in GF(p) and is represented as below.

$$A_2 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$\text{then } a_{11}^2 + a_{12}^2 = 1 \tag{3.9}$$

$$a_{11}a_{21} + a_{12}a_{22} = 0 \tag{3.10}$$

$$\text{and } a_{21}^2 + a_{22}^2 = 1 \tag{3.11}$$

If a_{11} and a_{22} are so selected such that

$$a_{11}^2 + a_{22}^2 = 1 \tag{3.12}$$

From equation (3.10)

$$a_{21} = \left(\frac{-1}{a_{11}}\right) a_{12}a_{22} \tag{3.13}$$

Substituting a_{21} from (3.13) in (3.11)

$$\frac{(a_{12}^2 a_{22}^2)}{(a_{11}^2)} + a_{22}^2 = 1$$

$$\text{or } \frac{(a_{12}^2 + a_{11}^2) * a_{22}^2}{a_{11}^2} = 1 \tag{3.14}$$

From equations (3.9) and (3.14)

$$a_{22} = \pm a_{11} \tag{3.15}$$

From equations (3.13) and (3.15)

$$a_{21} = -(\pm a_{12}) \tag{3.16}$$

For any value of m_1 and m_2

$$\text{If } x = \frac{m_1^2 - m_2^2}{m_1^2 + m_2^2} \text{ and } y = \frac{2m_1 m_2}{m_1^2 + m_2^2}$$

$$\text{then } x^2 + y^2 = 1$$

Using (3.12) and above equation, a_{11} and a_{12} can be easily obtained.

Example ($p = 251$) With $m_1 = 2$ and $m_2 = 1$, the following 2×2 orthonormal matrix A_2 can be obtained.

$$A_2 = \begin{bmatrix} 101 & 51 \\ 51 & 150 \end{bmatrix} \text{ or } \begin{bmatrix} 101 & 51 \\ 200 & 101 \end{bmatrix}$$

Higher order orthonormal matrices can be generated by the nine matrices (C_{11}, C_{12}, C_{21}, C_{22}, Z, D_{11}, D_{12}, D_{21}, D_{22}) as defined below.

$$C_{11} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \qquad C_{12} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$C_{21} = \begin{bmatrix} -1 & -1 \\ 0 & 0 \end{bmatrix} \qquad C_{22} = \begin{bmatrix} 0 & 0 \\ -1 & -1 \end{bmatrix}$$

$$Z = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$D_{11} = \begin{bmatrix} C_{11} & C_{12} \\ Z & Z \end{bmatrix} \qquad D_{12} = \begin{bmatrix} Z & Z \\ C_{11} & C_{12} \end{bmatrix}$$

$$D_{21} = \begin{bmatrix} C_{21} & C_{22} \\ Z & Z \end{bmatrix} \qquad D_{22} = \begin{bmatrix} Z & Z \\ C_{21} & C_{22} \end{bmatrix}$$

A 4 × 4 orthonormal matrix A_4 can be formulated as

$$A_4 = \begin{bmatrix} \frac{A_2C_{11}}{2} & \vdots & \frac{A_2C_{12}}{2} \\ \dots & \vdots & \dots \\ \frac{A_2C_{21}}{2} & \vdots & \frac{A_2C_{22}}{2} \end{bmatrix}$$

On the same principle, $2^m \times 2^m$ orthonormal matrix can be iteratively developed by using $2^{m-1} \times 2^{m-1}$ matrix and by replacing $C_{11} \leftarrow D_{11}$, $C_{12} \leftarrow D_{12}$, $C_{21} \leftarrow D_{21}$, $C_{22} \leftarrow D_{22}$ and $Z \leftarrow \begin{bmatrix} Z & Z \\ Z & Z \end{bmatrix}$

3.2.2 Method II

Let A be a partition matrix of size $2m \times 2m$ as shown below

$$A = \begin{bmatrix} A_{11} & \vdots & A_{12} \\ \dots & \vdots & \dots \\ A_{21} & \vdots & A_{22} \end{bmatrix} \tag{3.17}$$

where A_{11}, A_{12}, A_{22} are three different orthonormal matrices of size m × m and

$$A_{21} = -A_{22}A_{12}^T A_{11} \tag{3.18}$$

$$AA^T = \begin{bmatrix} A_{11} & \vdots & A_{12} \\ \dots & \vdots & \dots \\ A_{21} & \vdots & A_{22} \end{bmatrix} . \begin{bmatrix} A_{11}^T & \vdots & A_{21}^T \\ \dots & \vdots & \dots \\ A_{12}^T & \vdots & A_{22}^T \end{bmatrix} = 2I \tag{3.19}$$

If above equations 3.17, 3.18 and 3.19 are satisfied, then A will be orthogonal matrix. Thus by multiplying the matrix A by $\frac{1}{\sqrt{2}}$ one can get an orthonormal matrix.

Example (Modulo $p = 251$)

Let A_{11}, A_{12}, A_{21}, A_{22} be 4×4 orthonormal matrices.

$$A_{11} = \begin{bmatrix} 161 & 131 & 45 & 24 \\ 131 & 90 & 24 & 206 \\ 58 & 50 & 209 & 107 \\ 50 & 193 & 107 & 42 \end{bmatrix}$$

$$A_{12} = \begin{bmatrix} 245 & 123 & 65 & 38 \\ 123 & 6 & 38 & 186 \\ 189 & 90 & 20 & 61 \\ 90 & 62 & 61 & 231 \end{bmatrix}$$

$$A_{22} = \begin{bmatrix} 99 & 120 & 159 & 79 \\ 120 & 152 & 79 & 92 \\ 66 & 178 & 20 & 61 \\ 178 & 185 & 61 & 231 \end{bmatrix}$$

A_{21} is obtained by substituting values of A_{11}, A_{12} and A_{22} in (3.18)

$$A_{21} = \begin{bmatrix} 51 & 71 & 212 & 80 \\ 71 & 200 & 80 & 39 \\ 244 & 200 & 239 & 138 \\ 200 & 7 & 138 & 12 \end{bmatrix}$$

$$\text{So, } A = \begin{bmatrix} 161 & 131 & 45 & 24 & 245 & 123 & 65 & 38 \\ 131 & 90 & 24 & 206 & 123 & 6 & 38 & 186 \\ 58 & 50 & 209 & 107 & 189 & 90 & 20 & 61 \\ 50 & 193 & 107 & 42 & 90 & 62 & 61 & 231 \\ 51 & 71 & 212 & 80 & 99 & 120 & 159 & 79 \\ 71 & 200 & 80 & 39 & 120 & 152 & 79 & 92 \\ 244 & 200 & 239 & 138 & 66 & 178 & 20 & 61 \\ 200 & 7 & 138 & 12 & 178 & 185 & 61 & 231 \end{bmatrix}$$

This 8×8 matrix A is orthogonal and satisfies the relationship (3.19)

3.2.3 Method III

Let A_{11}, A_{12} and A_{22} be orthonormal matrices. The following matrix A will be orthonormal,

$$A = \begin{bmatrix} n_1 A_{11} & n_2 A_{12} \\ n_2 A_{21} & n_1 A_{22} \end{bmatrix} \tag{3.20}$$

$$\text{if } A_{21} = -A_{22} A_{12}^T A_{11} \text{ and } n_1^2 + n_2^2 = 1.$$

n_1 and n_2 are integers which can be determined by the following relation

$$n_1 = \frac{m_1^2 - m_2^2}{m_1^2 + m_2^2},\ n_2 = \frac{2 m_1 m_2}{m_1^2 + m_2^2}$$

where, m_1 and m_2 are also integers.

3.3 Proposed method of generating Orthonormal Matrices in GF(2^n)

If $f(x)$ is a polynomial in GF(2^n), a 2×2 orthonormal matrix A in GF(2^n) can be of the form

$$A = \begin{bmatrix} f(x) & f(x)+1 \\ f(x)+1 & f(x) \end{bmatrix} \tag{3.21}$$

Then a generalised orthonormal matrix $F_{2m,2m}(x)$ can be generated using orthonormal matrix $F_{m,m}(x)$ as shown below.

$$F_{2m,2m}(x) = \begin{bmatrix} F_{mm}(x) & F_{mm}(x)+I \\ F_{mm}(x)+I & F_{mm}(x) \end{bmatrix} \tag{3.22}$$

3.4 Proposed method of generating Orthonormal matrix in GF(p^n)

Let A be $n \times n$ orthonormal matrix in GF(p^n).

$$A = \begin{bmatrix} a_{11}(x) & a_{12}(x) \\ a_{21}(x) & a_{22}(x) \end{bmatrix} \tag{3.23}$$

$$\text{then, } A.A^T = I \Rightarrow a_{11}^2(x) + a_{12}^2(x) = 1, \tag{3.24}$$

$$a_{11}(x)a_{21}(x) + a_{12}(x)a_{22}(x) = 0, \tag{3.25}$$

$$\text{and } a_{21}^2(x) + a_{22}^2(x) = 1 \tag{3.26}$$

Let $m_1(x)$ and $m_2(x)$ be two elements in GF(p^n)

$$\text{then } a_{11}(x) = \frac{(m_1^2(x) - m_2^2(x))}{(m_1^2(x) + m_2^2(x))} \tag{3.27}$$

$$\text{and } a_{12}(x) = \frac{2m_1(x)m_2(x)}{(m_1^2(x) + m_1^2(x))} \tag{3.28}$$

Substituting $a_{11}(x)$ and $a_{12}(x)$ in equation (3.25) and (3.26), $a_{21}(x)$ and $a_{22}(x)$ can be obtained. Now a $2^m \times 2^m$ matrix B can be generated by evaluating three

$(2^{m-1} \times 2^{m-1})$ orthonormal matrices $A_{11}(x), A_{12}(x)$ and $A_{22}(x)$ as per the following method.

$$\text{Let } B = \begin{bmatrix} b(x)A_{11}(x) & c(x)A_{12}(x) \\ c(x)A_{21}(x) & b(x)A_{11}(x) \end{bmatrix} \tag{3.29}$$

where, b(x) and c(x) are elements in GF(p^n).

B will be orthonormal matrix, if

$$b^2(x) + c^2(x) = 1 \tag{3.30}$$

$$\text{and } A_{21}(x) = -A_{22}(x)A_{12}^T(x)A_{11}(x) \tag{3.31}$$

3.5 Proposed algorithm of Encryption and Decryption

(a) Encryption

Step 1. Generate an orthonormal matrix 'A' by any one of the methods narrated in sections 3.2 through 3.4.

Step 2. Select 'n' degree polynomials with non-zero roots arbitrarily.

Step 3. Generate a matrix 'B' with eigen factors derived from Step 2 and using the method discussed in section 2.3.

Step 4. Determine the key matrix 'C' for encryption by the relation $C = A.B.A^T$.

Step 5. Divide the image into 8 × 8 blocks.

Step 6. Encrypt each block by the key matrix.

Step 7. Combine these blocks to form the encrypted image.

(a) Decryption

Step 1. Divide the encrypted image into 8 × 8 blocks.

Step 2. Generate decryption matrix 'D' by the relation $D = A.B^{-1}.A^T$ as discussed in section 2.3.2.

Step 3. Decrypt each block using the decryption matrix.

Step 4. Form the decrypted image by combining these blocks.

3.6 Cryptanalysis

Number of 2×2 orthonormal matrices that exist in $GF(p)$ equals to $2(p-1)^2 - p$. When the dimension of the matrix is increased and p is taken as a large number, the number of orthonormal matrices increases exponentially. Therefore, it becomes very difficult for the intruder to guess the key-matrix by brute force attack. The proposed algorithm uses asymmetric key technique for encryption where the public key equals to $A.B.A^T$. Here, A is the orthonormal matrix and B is generated as discussed in section 2.3. Decryption is easy if the private keys A and B are known as inverse of these keys can be easily determined. But, the intruder has to find the inverse of the public key using standard matrix inversion methods.

It is also difficult to predict the key by cipher-text-attack as the image is completely scrambled leaving no trace of the original image. The statistical property of the image is also lost which is demonstrated in the histogram of the encrypted image.

From the above analysis, it is evident that one cannot find the key by plain-text/cipher-text attack. Moreover image encryption by block transformation technique increases the difficulty level.

Above cryptanalysis also holds good for encryption in GF(2^8) and GF(13^2).

3.7 Results

The proposed algorithm for encryption and decryption is validated using simulation technique. In the simulation studies, images of different resolution i.e. 256 $\times$ 256 and 512 $\times$ 512 pixels with 8 bit encoding for each pixel is considered.

3.7.1 Simulation result of encryption in GF(p) ($p = 251$)

Let A be an orthonormal matrix selected randomly in GF(p).

$$A = \begin{bmatrix} 56 & 242 & 223 & 52 & 143 & 206 & 166 & 182 \\ 242 & 195 & 52 & 28 & 206 & 108 & 182 & 85 \\ 42 & 192 & 160 & 190 & 139 & 114 & 109 & 94 \\ 192 & 209 & 190 & 91 & 114 & 112 & 94 & 142 \\ 165 & 23 & 51 & 185 & 89 & 9 & 219 & 213 \\ 23 & 86 & 185 & 200 & 9 & 162 & 213 & 32 \\ 125 & 86 & 35 & 225 & 143 & 51 & 127 & 174 \\ 86 & 126 & 225 & 216 & 51 & 108 & 174 & 124 \end{bmatrix}$$

The private key matrix B (in GF(p)) has been generated by taking eigen factors $x - 31$, $x - 7$, $x - 191$, $x^3 + x^2 + x - 8$, $x - 47$ and $x - 127$ as defined in Section

2.3. So, the private key B can be presented as

$$B = \begin{bmatrix} 31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 191 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 47 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 127 \end{bmatrix}$$

The public key matrix C (in GF(p)) is generated by using the relation $C = A.B.A^T$

$$C = \begin{bmatrix} 3 & 208 & 49 & 179 & 68 & 73 & 67 & 17 \\ 18 & 41 & 2 & 22 & 134 & 176 & 186 & 36 \\ 170 & 208 & 156 & 39 & 8 & 139 & 54 & 96 \\ 16 & 217 & 178 & 103 & 139 & 3 & 65 & 179 \\ 91 & 118 & 236 & 100 & 183 & 10 & 172 & 156 \\ 223 & 6 & 40 & 130 & 84 & 238 & 78 & 211 \\ 234 & 212 & 237 & 83 & 168 & 93 & 105 & 226 \\ 45 & 121 & 240 & 208 & 2 & 66 & 204 & 75 \end{bmatrix}$$

The inverse of public key D (in GF(p)) is generated by using the relation $D = A.B^{-1}.A^T$

$$D = \begin{bmatrix} 70 & 113 & 213 & 19 & 221 & 122 & 50 & 248 \\ 39 & 215 & 164 & 91 & 205 & 3 & 192 & 7 \\ 238 & 97 & 48 & 231 & 115 & 96 & 55 & 188 \\ 0 & 185 & 152 & 36 & 24 & 160 & 189 & 114 \\ 217 & 111 & 225 & 239 & 117 & 229 & 232 & 26 \\ 6 & 80 & 97 & 124 & 177 & 202 & 14 & 111 \\ 127 & 121 & 206 & 177 & 110 & 244 & 99 & 162 \\ 206 & 103 & 24 & 126 & 227 & 211 & 21 & 187 \end{bmatrix}$$

256×256 cameraman image and 512×512 baboon image shown in Figures 2.1 and 3.1 respectively are considered to carry out the simulation. Their histograms are presented in Figures 2.2 and 3.2 respectively. Since encryption is done in GF(p) field with $p = 251$, pixels of original image beyond 250 are rounded to 250. The images are rounded to 250 pixels and their histograms are shown in Figures 2.3, 3.3, 2.4 and 3.4 respectively.

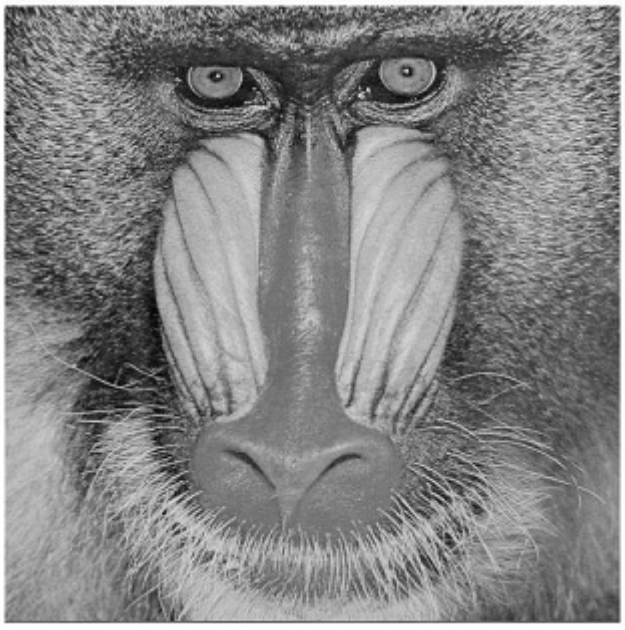

Figure 3.1: Original baboon image used for encryption

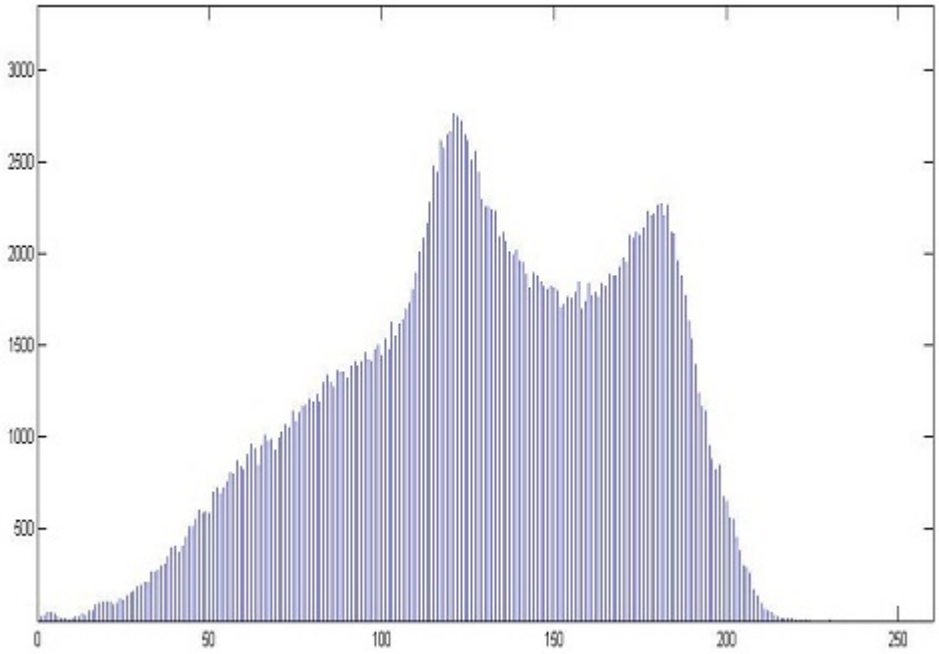

Figure 3.2: Histogram of original baboon image

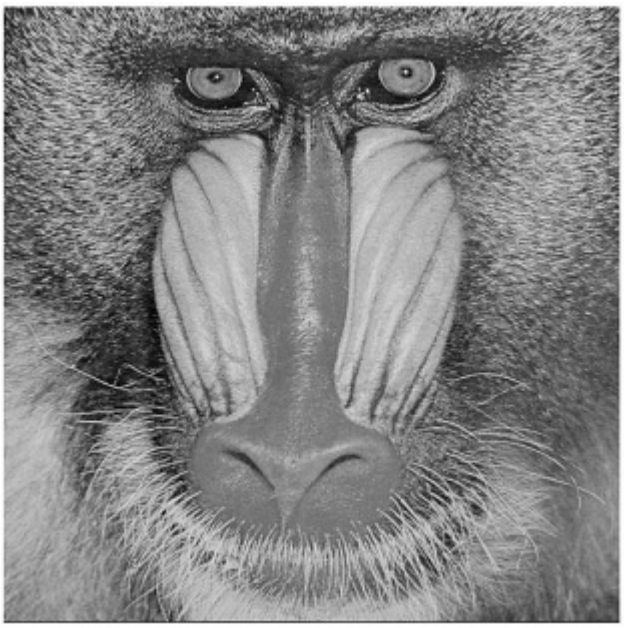

Figure 3.3: Baboon image with pixels ≤ 250

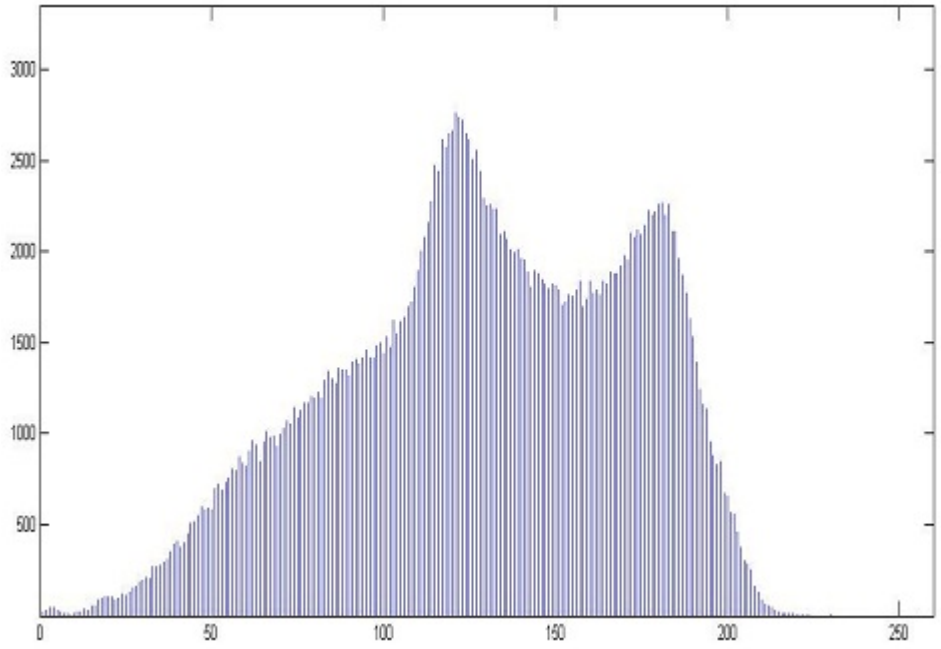

Figure 3.4: Histogram of baboon image with pixels ≤ 250

The encryption algorithm proposed in section 3.5 is used to encrypt the cameraman and baboon images which are rounded to 250 pixels. After encryption, the images of cameraman and baboon and their corresponding histograms are shown in Figures 3.5, 3.6 and 3.7 respectively. On comparing the encrypted images with the original images, it can be seen that the encrypted images and their histograms do not bear any resemblance with the original images and their histograms.

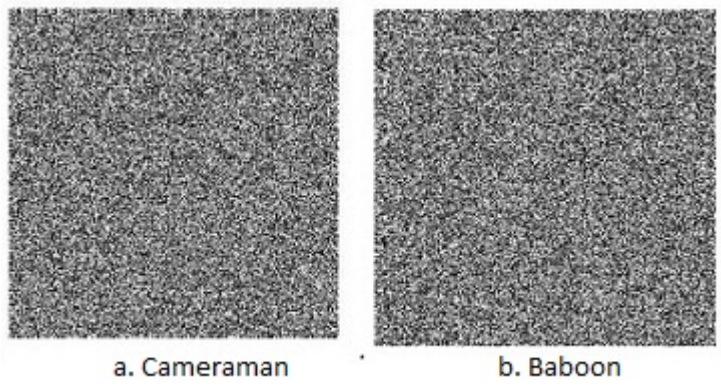

Figure 3.5: Encrypted image of cameraman and baboon in $GF(p)$ (where $p = 251$) using encryption algorithm in section 3.5

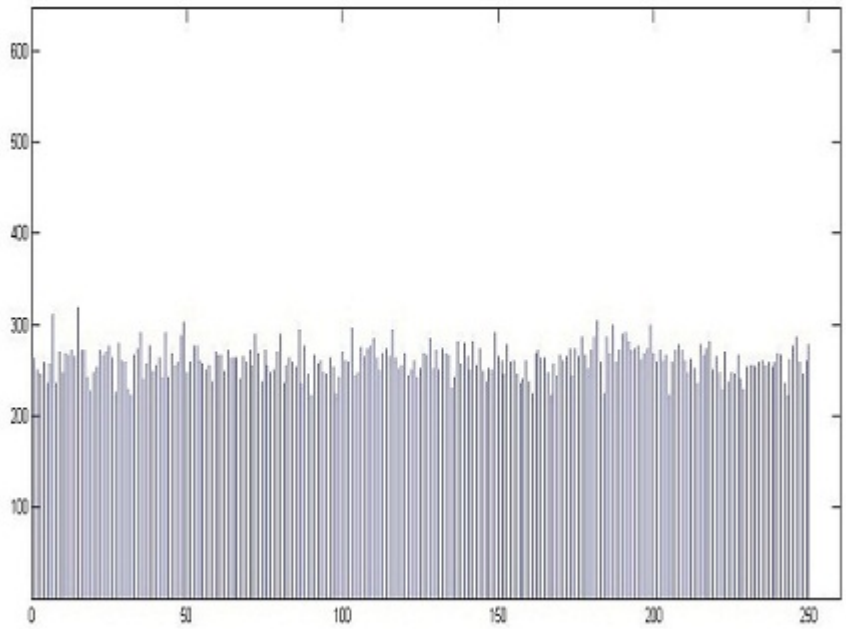

Figure 3.6: Histogram of encrypted image of cameraman in GF(p) (where $p = 251$) using encryption algorithm in section 3.5

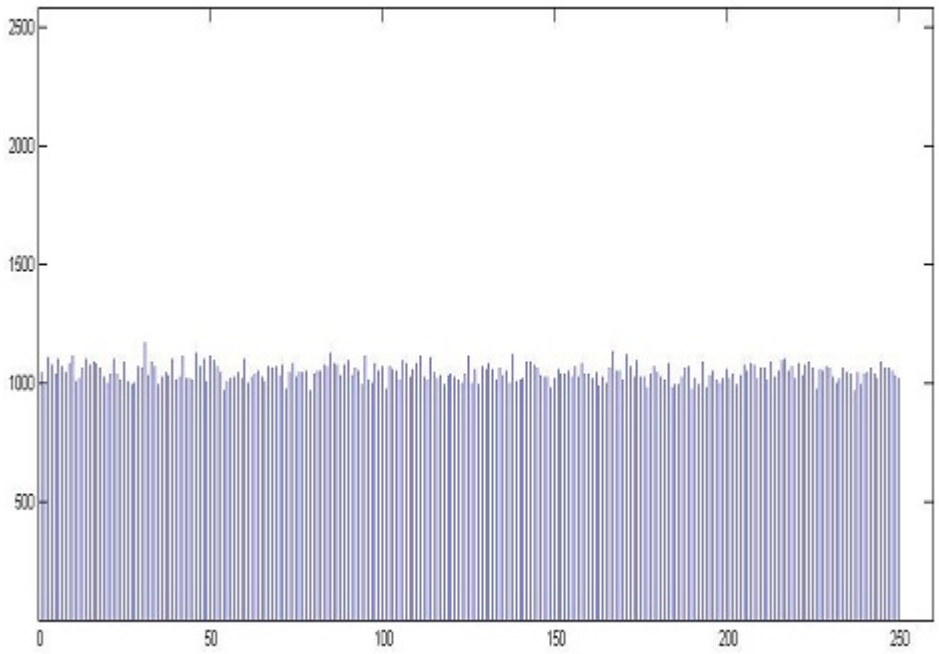

Figure 3.7: Histogram of encrypted image of baboon in GF(p) (where $p = 251$) using encryption algorithm in section 3.5

The encrypted images are decrypted by using the decryption algorithm presented in section 3.5. The decrypted images and their corresponding histograms are shown in Figures 3.8, 3.9 and 3.10 respectively which are same as the original images(rounded to 250 pixels) and their corresponding histograms.

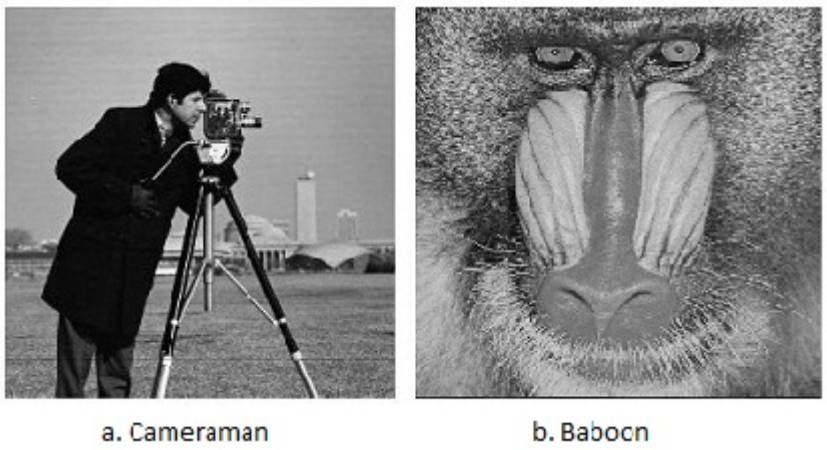

Figure 3.8: Decrypted image of cameraman and baboon in $GF(p)$ (where $p = 251$) using decryption algorithm in section 3.5

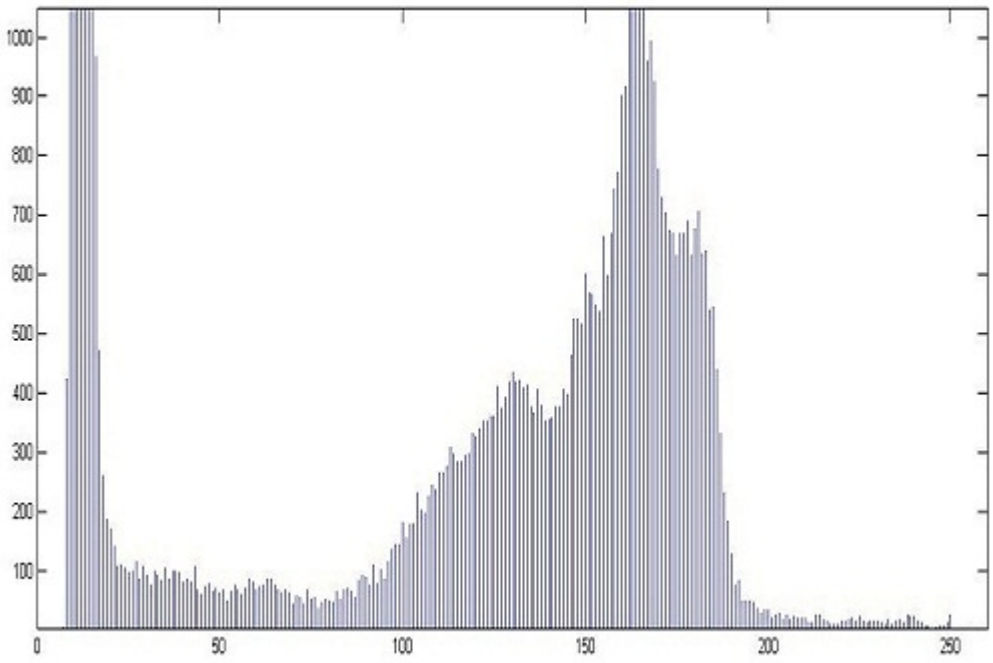

Figure 3.9: Histogram of decrypted image of cameraman in $GF(p)$ (where $p = 251$) using decryption algorithm in section 3.5

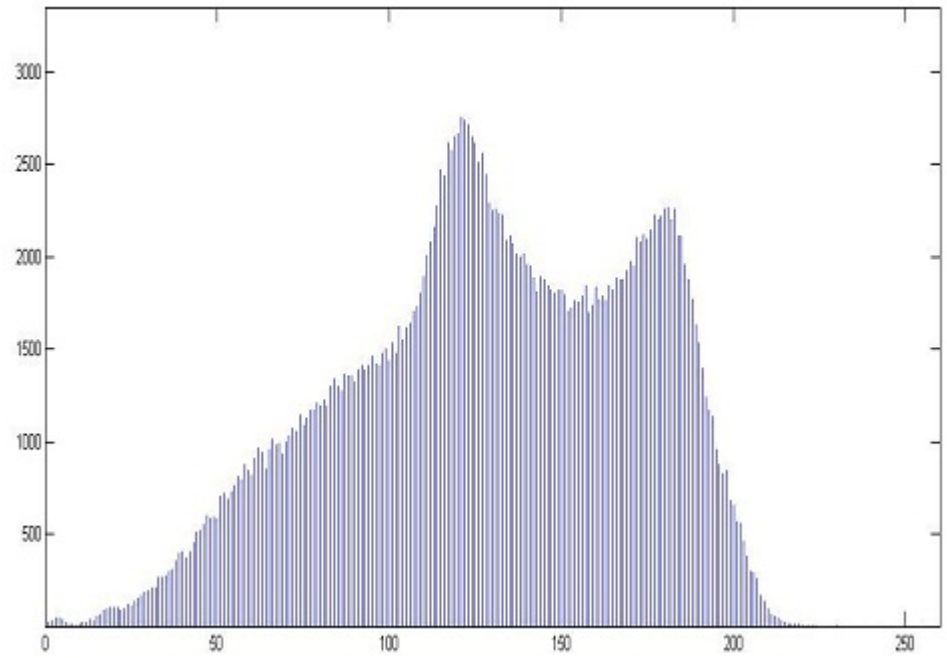

Figure 3.10: Histogram of decrypted image of baboon in $GF(p)$ (where $p = 251$) using decryption algorithm in section 3.5

3.7.2 Simulation result of encryption over $GF(2^8)$

The primitive polynomial for 8-bit data encryption in GF(2^8) is $D^8 + D^4 + D^3 + D^2 + 1 = 285$.

Let A be a self-invertible matrix selected randomly in GF(2^8) whose decimal equivalent is given here.

$$A = \begin{bmatrix} 166 & 203 & 44 & 180 & 83 & 235 & 22 & 90 \\ 203 & 166 & 180 & 44 & 235 & 83 & 90 & 22 \\ 229 & 125 & 228 & 137 & 252 & 176 & 114 & 202 \\ 125 & 229 & 137 & 228 & 176 & 252 & 202 & 114 \\ 164 & 28 & 180 & 248 & 85 & 56 & 117 & 237 \\ 28 & 164 & 248 & 180 & 56 & 85 & 237 & 117 \\ 76 & 0 & 83 & 235 & 152 & 0 & 166 & 203 \\ 0 & 76 & 235 & 83 & 0 & 152 & 203 & 166 \end{bmatrix}$$

The private key matrix B (in $GF(2^8)$) has been generated by taking eigen factors $x-(D+1)$, $x-(D^2+D+1)$, $x-(D^3+D^2+1)$, $x-(D^4+D^3+1)$, $x-(D^4+D^3+D^2+D+1)$ and $x^3-x^2-x-D^3$ as defined in Section 2.3. So, decimal equivalent of the private key B can be presented as

$$B = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 31 \end{bmatrix}$$

The public key matrix C (in $GF(2^8)$) is generated by using the relation $C = A.B.A^T$. The decimal equivalent of C is given as below.

$$C = \begin{bmatrix} 141 & 159 & 226 & 82 & 232 & 92 & 119 & 84 \\ 214 & 54 & 9 & 92 & 15 & 196 & 75 & 195 \\ 103 & 252 & 132 & 196 & 95 & 131 & 209 & 155 \\ 72 & 148 & 42 & 147 & 147 & 180 & 163 & 144 \\ 27 & 21 & 2 & 16 & 239 & 166 & 35 & 148 \\ 177 & 248 & 99 & 5 & 79 & 36 & 0 & 127 \\ 241 & 21 & 229 & 62 & 157 & 19 & 181 & 76 \\ 96 & 168 & 21 & 143 & 215 & 51 & 3 & 220 \end{bmatrix}$$

The inverse of public key D (in $GF(2^8)$) is generated by using the relation $D = A.B^{-1}.A^T$. The decimal equivalent of D is given as below.

$$D = \begin{bmatrix} 54 & 199 & 239 & 180 & 50 & 52 & 214 & 186 \\ 111 & 236 & 23 & 197 & 240 & 159 & 213 & 117 \\ 251 & 154 & 127 & 181 & 224 & 50 & 129 & 134 \\ 177 & 234 & 202 & 147 & 26 & 126 & 19 & 119 \\ 181 & 224 & 255 & 146 & 154 & 231 & 221 & 79 \\ 17 & 78 & 252 & 127 & 45 & 15 & 209 & 189 \\ 16 & 95 & 52 & 119 & 241 & 222 & 77 & 79 \\ 64 & 55 & 16 & 83 & 95 & 61 & 254 & 83 \end{bmatrix}$$

The encryption algorithm proposed in section 3.5 is used to encrypt the cameraman and baboon images in $GF(2^8)$. The encrypted images and their corresponding histograms are shown in Figures 3.11, 3.12 and 3.13 respectively. On comparing the encrypted images with the original images, it can be seen that the encrypted images and their corresponding histograms do not bear any resemblance with the original images and their histograms.

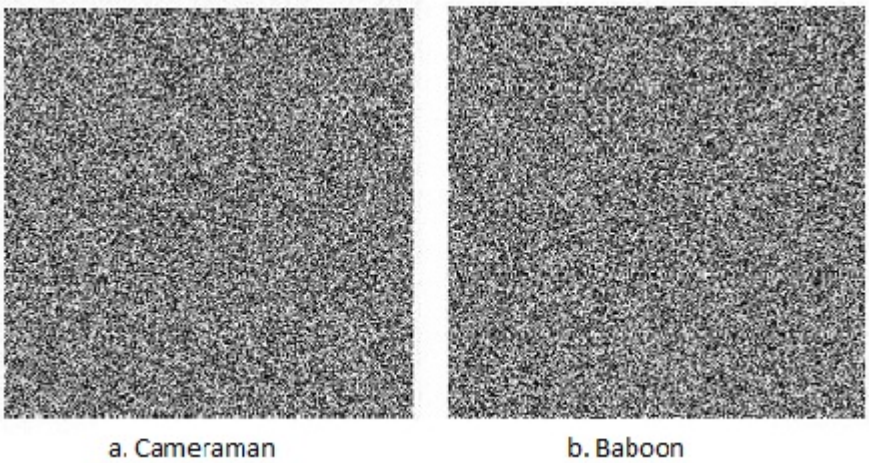

Figure 3.11: Encrypted images of cameraman and baboon in $GF(2^8)$ using encryption algorithm in section 3.5

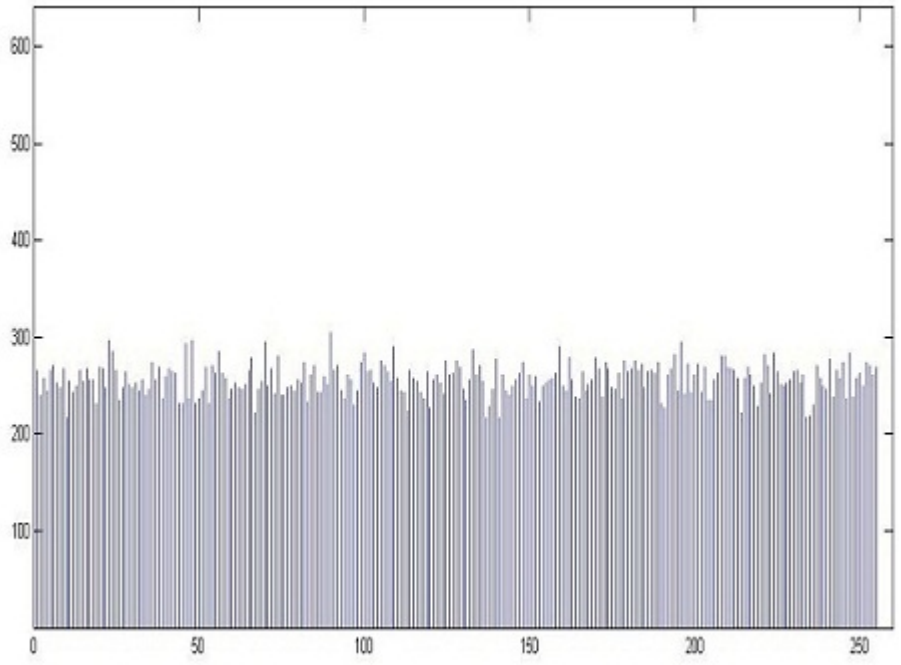

Figure 3.12: Histogram of encrypted image of cameraman in $GF(2^8)$ using encryption algorithm in section 3.5

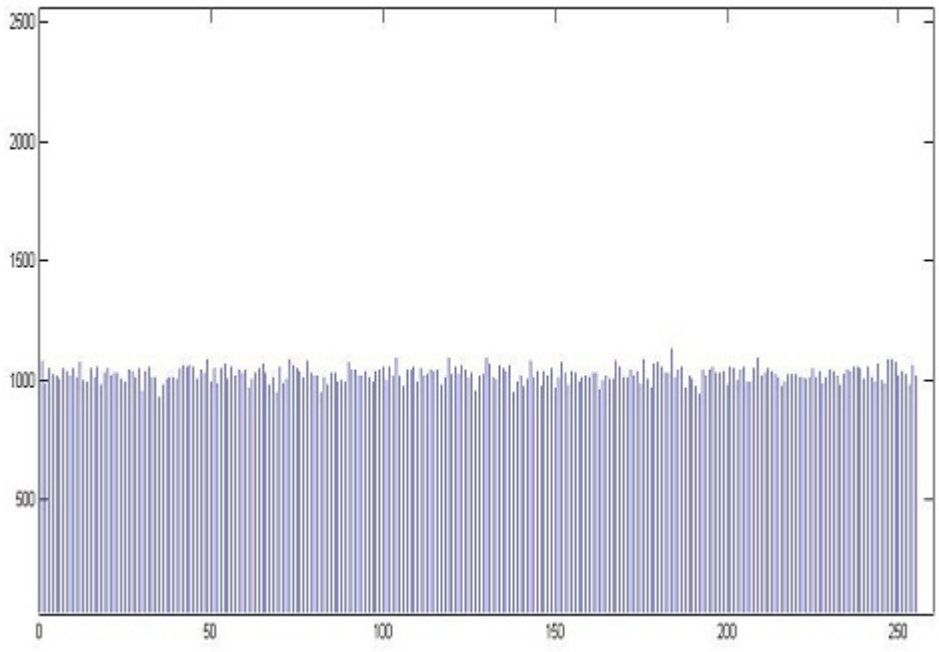

Figure 3.13: Histogram of encrypted image of baboon in $GF(2^8)$ using encryption algorithm in section 3.5

The encrypted images are decrypted by using the decryption algorithm presented in section 3.5. The decrypted images and their histograms are shown in Figures 3.14, 3.15 and 3.16 respectively which are same as the original images and their histograms.

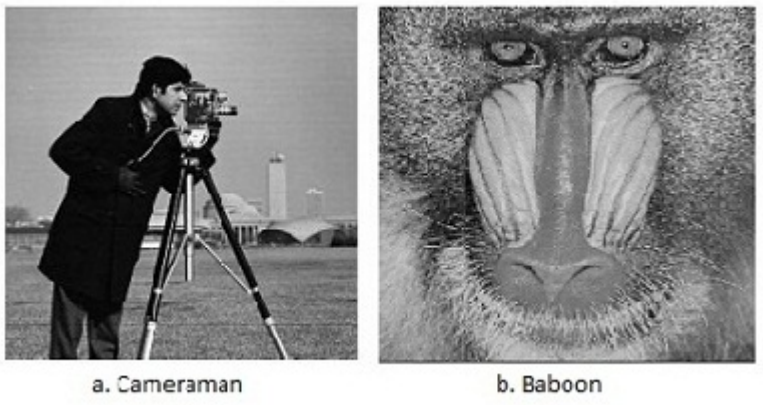

Figure 3.14: Decrypted images of cameraman and baboon in $GF(2^8)$ using decryption algorithm in section 3.5

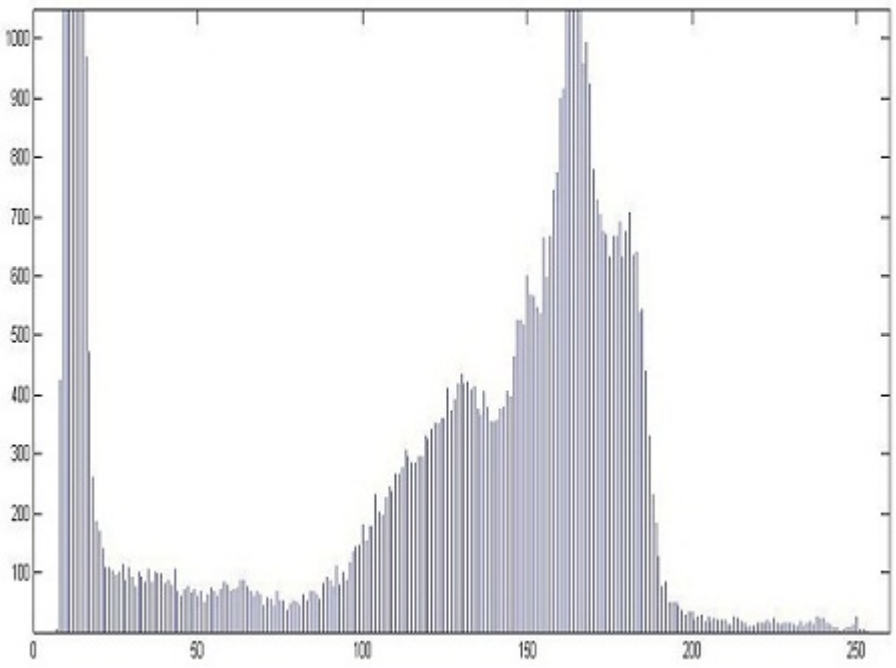

Figure 3.15: Histogram of decrypted image of cameraman in $GF(2^8)$ using decryption algorithm in section 3.5

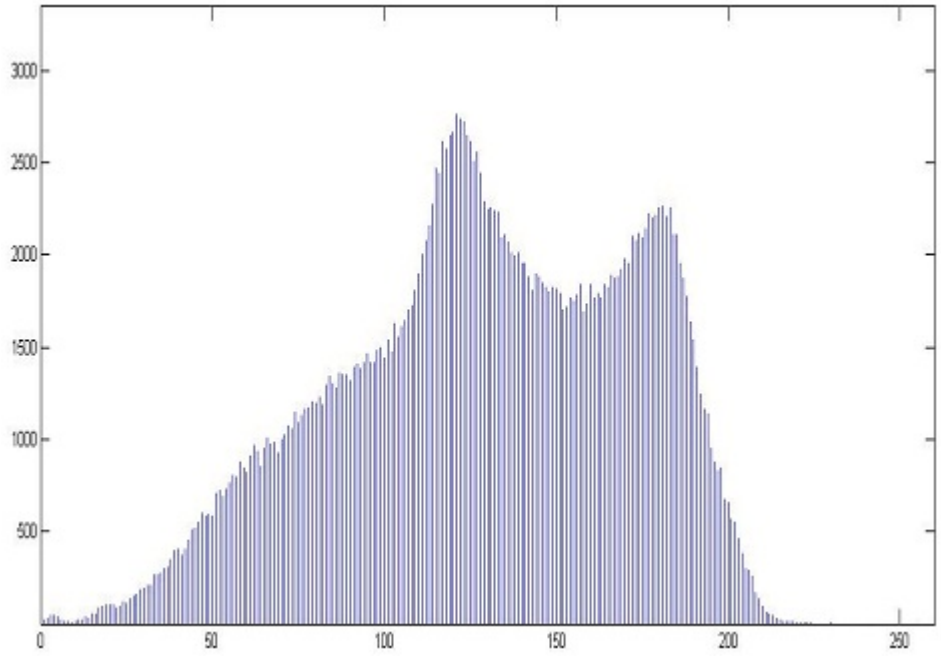

Figure 3.16: Histogram of decrypted image of baboon in $GF(2^8)$ using decryption algorithm in section 3.5

3.7.3 Simulation result of encryption over $GF(p^n)$

Encryption of cameraman and baboon images in GF(2^8) is presented in previous section. In this section, encryption in GF(p^n) is presented considering $p = 13$ and $n = 2$ and the primitive polynomial over GF(p^n) $D^2 + D^1 + 2 = 184$ has been considered for encryption. Let A be a self-invertible matrix selected randomly in GF(p^n) whose decimal equivalent is given here.

$$A = \begin{bmatrix} 101 & 126 & 71 & 14 & 104 & 47 & 151 & 140 \\ 56 & 101 & 168 & 71 & 135 & 104 & 42 & 151 \\ 162 & 84 & 32 & 18 & 47 & 112 & 19 & 73 \\ 98 & 162 & 164 & 32 & 70 & 47 & 109 & 19 \\ 145 & 105 & 9 & 158 & 101 & 126 & 71 & 14 \\ 77 & 145 & 24 & 9 & 56 & 101 & 168 & 71 \\ 134 & 103 & 167 & 21 & 162 & 84 & 32 & 18 \\ 79 & 134 & 161 & 167 & 98 & 162 & 164 & 32 \end{bmatrix}$$

The private key matrix B (in $GF(p^n)$) has been generated by taking eigen factors $x-1$, $x-(10D+1)$, $x-(9D+1)$, x^3-x^2-x-8, $x-(4D+1)$ and $x-(6D+1)$ as defined in Section 2.3. So, decimal equivalent of the private key B can be presented as

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 131 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 118 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 53 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 79 \end{bmatrix}$$

The public key matrix C (in $GF(p^n)$) is generated by using the relation $C = A.B.A^T$. The decimal equivalent of C is given as below.

$$C = \begin{bmatrix} 68 & 119 & 123 & 100 & 63 & 113 & 85 & 16 \\ 33 & 44 & 164 & 99 & 147 & 94 & 32 & 48 \\ 52 & 134 & 1 & 54 & 137 & 107 & 7 & 152 \\ 42 & 62 & 34 & 149 & 68 & 4 & 123 & 70 \\ 49 & 118 & 47 & 93 & 132 & 35 & 85 & 135 \\ 132 & 53 & 35 & 102 & 154 & 160 & 56 & 119 \\ 61 & 152 & 14 & 125 & 19 & 71 & 90 & 13 \\ 93 & 9 & 51 & 15 & 26 & 32 & 25 & 116 \end{bmatrix}$$

The inverse of public key D (in GF(p^n)) is generated by using the relation $D = A.B^{-1}.A^T$. The decimal equivalent of D is given as below.

$$D = \begin{bmatrix} 104 & 119 & 115 & 31 & 0 & 49 & 135 & 90 \\ 83 & 131 & 76 & 162 & 148 & 22 & 123 & 108 \\ 66 & 23 & 141 & 6 & 46 & 167 & 15 & 22 \\ 140 & 28 & 21 & 18 & 158 & 90 & 50 & 91 \\ 148 & 87 & 47 & 43 & 60 & 49 & 1 & 52 \\ 147 & 141 & 122 & 20 & 61 & 68 & 19 & 160 \\ 138 & 151 & 22 & 163 & 104 & 158 & 35 & 137 \\ 15 & 138 & 150 & 50 & 26 & 23 & 152 & 82 \end{bmatrix}$$

Since encryption is done in GF(p^n) with $p = 13$ and $n = 2$, pixels of original image beyond 168 are rounded to 168. The image rounded to 168 pixels and its histogram are shown in Figures 3.17, 3.18 and 3.19 respectively.

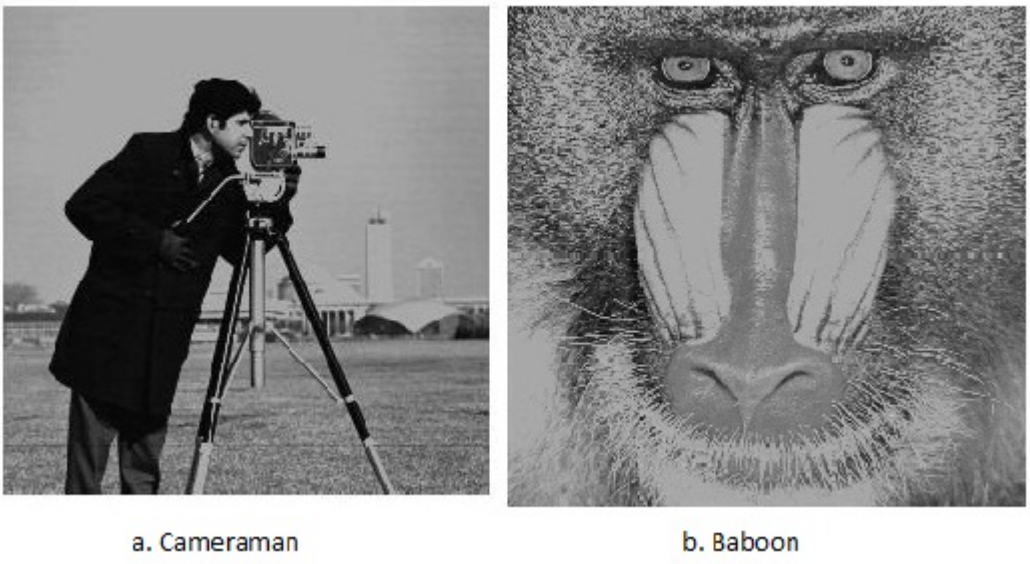

Figure 3.17: Cameraman and baboon images with pixels ≤ 168

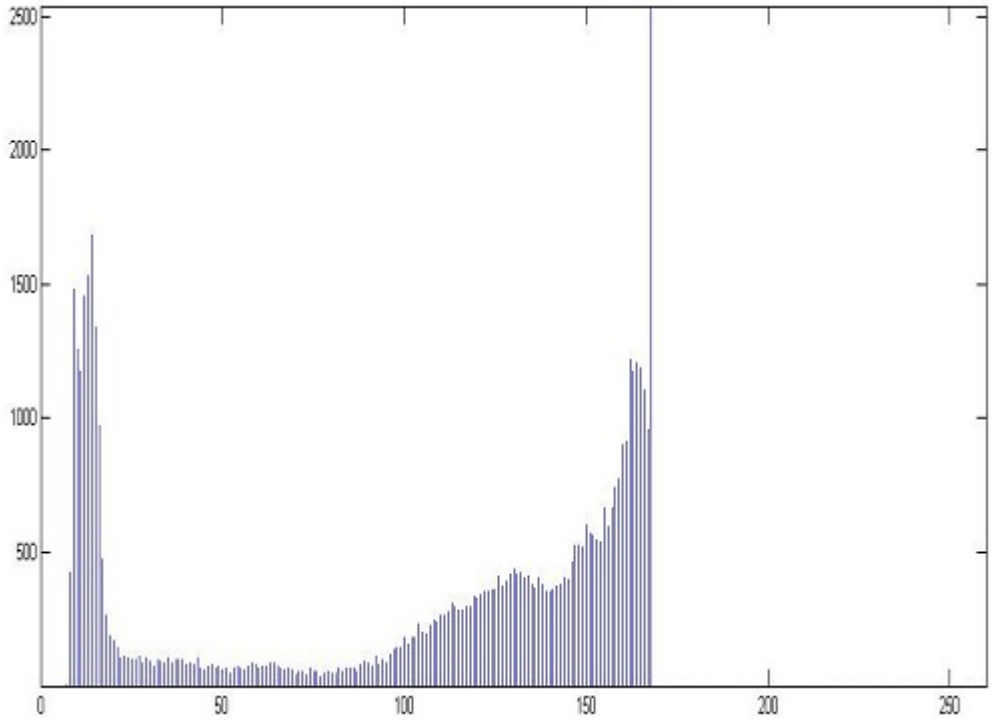

Figure 3.18: Histogram of cameraman image with pixels ≤ 168

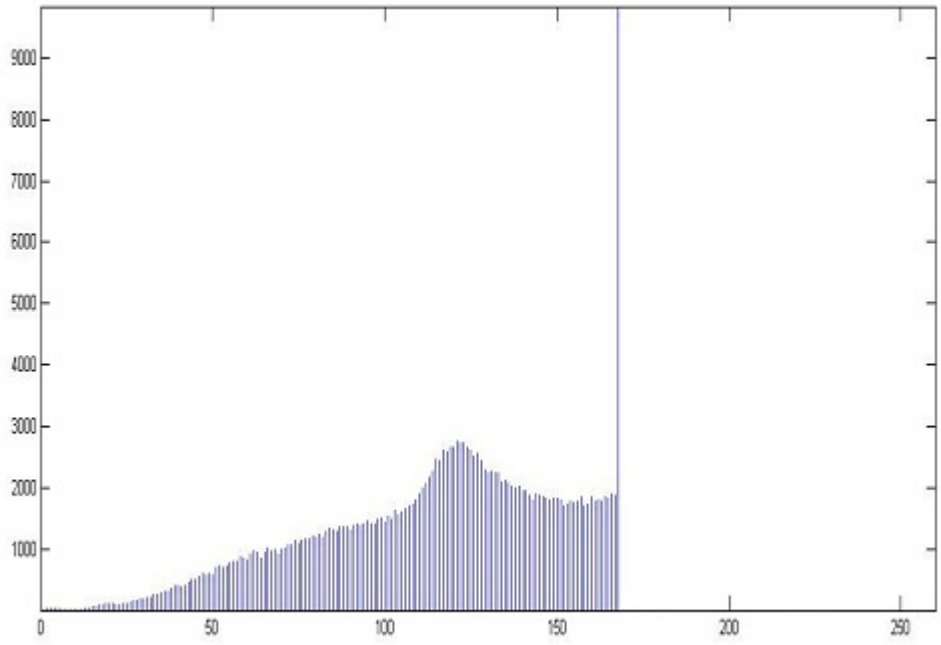

Figure 3.19: Histogram of baboon image with pixels ≤ 168

The encryption algorithm proposed in section 3.5 is used to encrypt the cameraman and baboon images which are rounded to 168 pixels. The encrypted images and their corresponding histograms are shown in Figures 3.20, 3.21 and 3.22 respectively. On comparing the encrypted images with the original images, it can be seen that the encrypted images and their histograms do not bear any resemblance with the original images and their histograms.

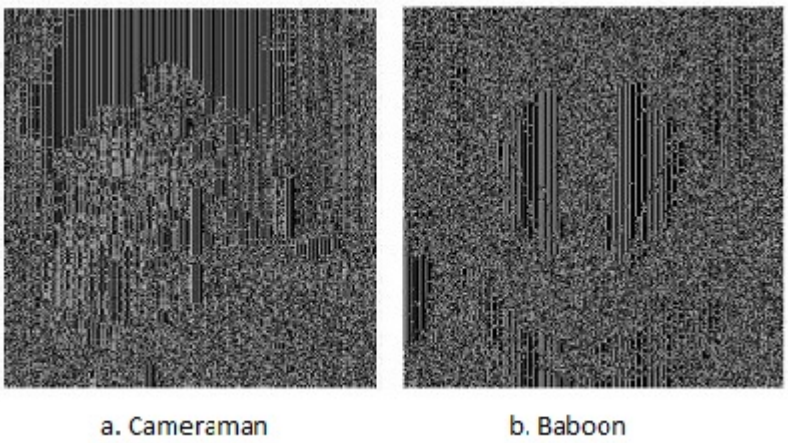

Figure 3.20: Encrypted images of cameraman and baboon in GF(13^2) using encryption algorithm in section 3.5

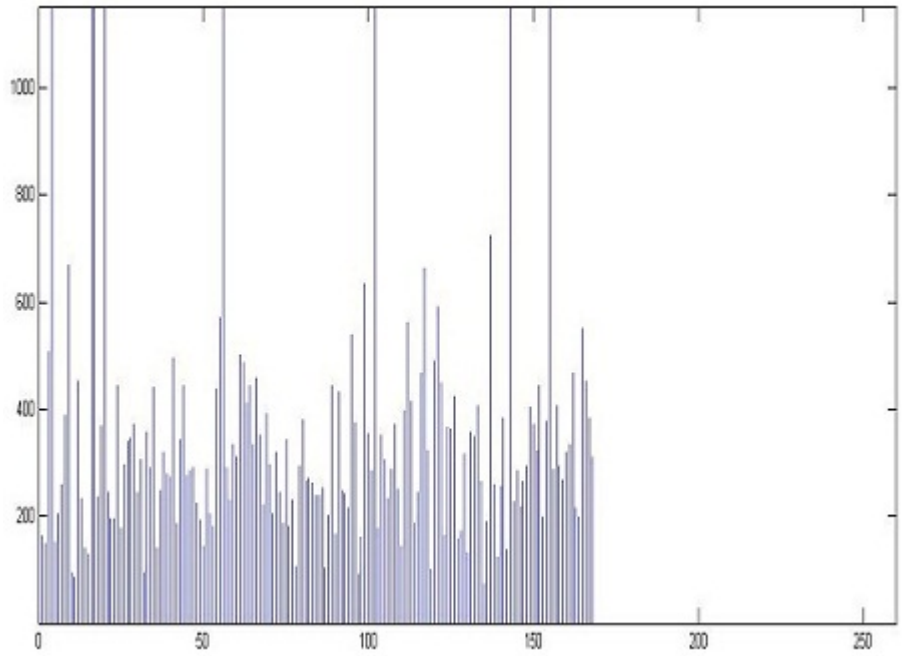

Figure 3.21: Histogram of encrypted image of cameraman in GF(13^2) using encryption algorithm in section 3.5

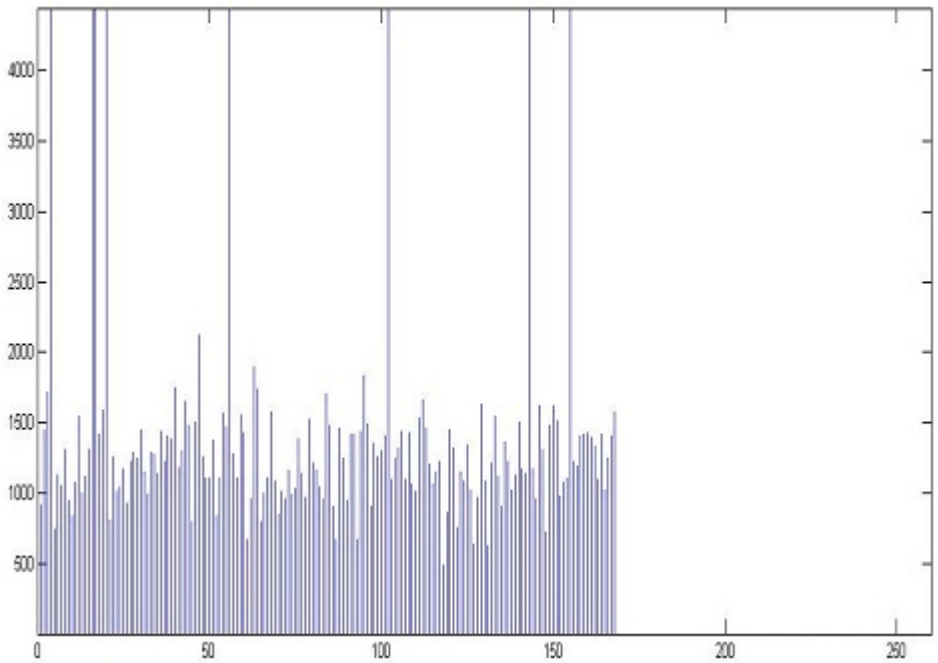

Figure 3.22: Histogram of encrypted image of baboon in GF(13^2) using encryption algorithm in section 3.5

The encrypted image is decrypted by using the decryption algorithm presented

in section 3.5. The decrypted images and their histograms are shown in Figures 3.23, 3.24 and 3.25 respectively which are same as the original images (rounded to 250 pixels) and their histograms.

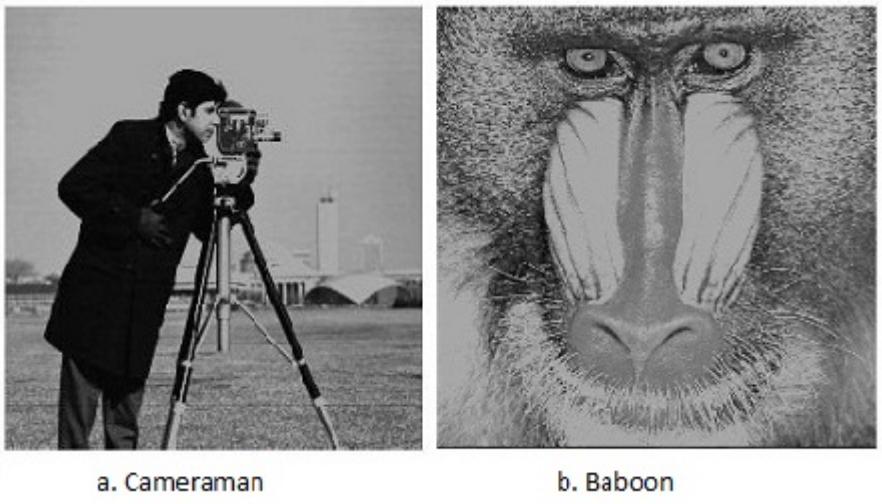

Figure 3.23: Decrypted images of cameraman and baboon in GF(13^2) using decryption algorithm in section 3.5

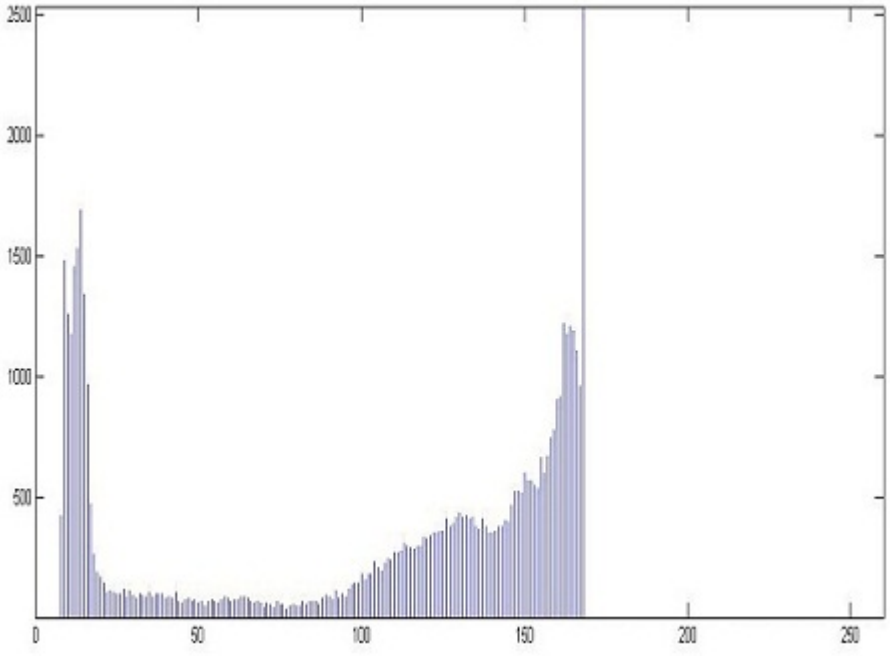

Figure 3.24: Histogram of decrypted image of cameraman in GF(13^2) using decryption algorithm in section 3.5

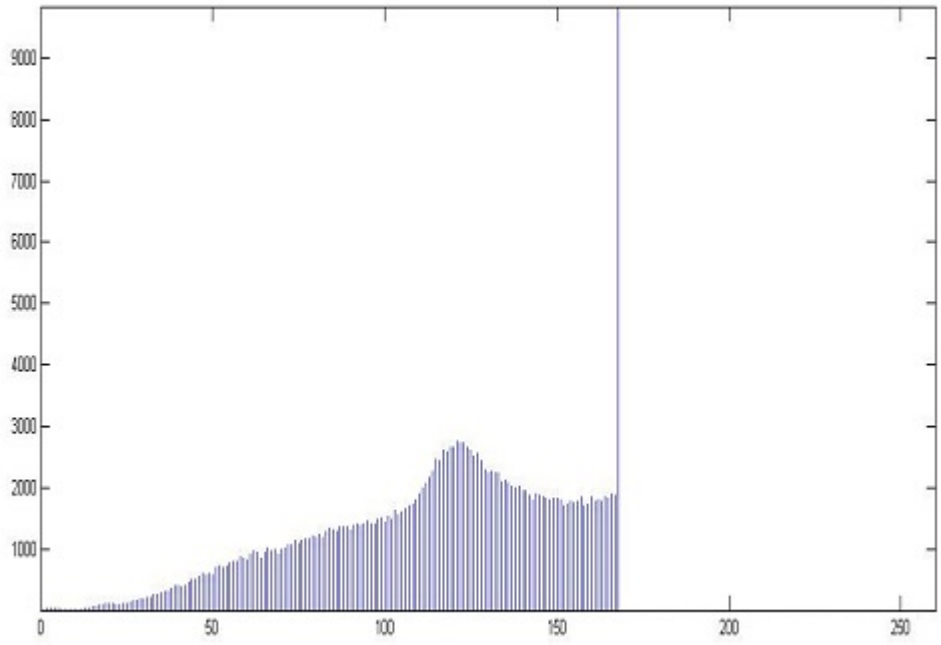

Figure 3.25: Histogram of decrypted image of baboon in GF(13^2) using decryption algorithm in section 3.5

The proposed algorithm is also tested with other images. Some of the simulation results are shown in Figure 3.26, where Figure 3.26(a,b,c) represent the original images of Cell, Goldhill and Pepper. The corresponding histograms of the original images are shown in Figure 3.26(d,e,f). The encryption algorithm proposed in section 3.5 is used for encrypting the images. The encrypted images are shown in Figure 3.26(g,h,i) and their histograms are represented in Figure 3.26(j,k,l).

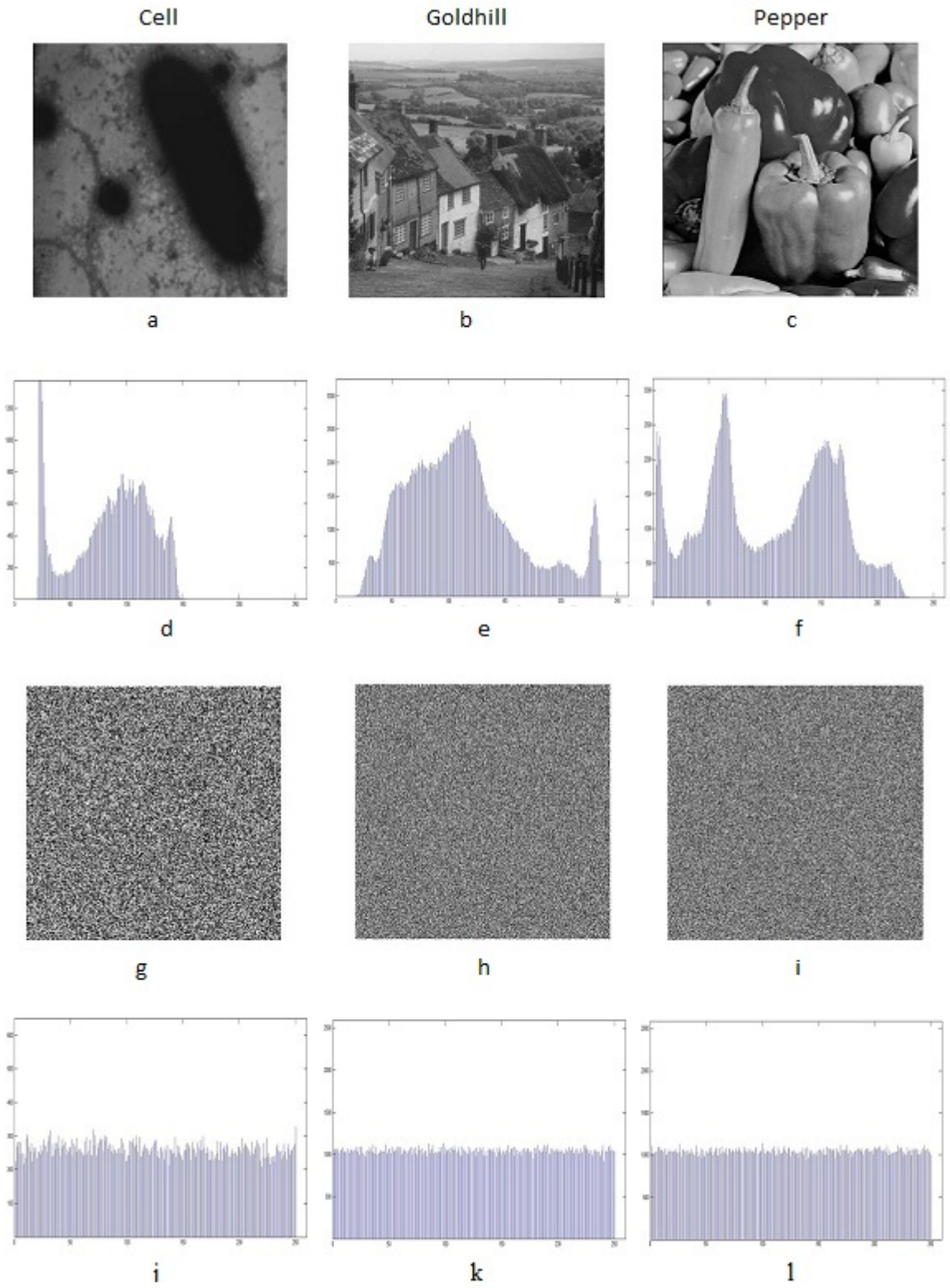

Figure 3.26: Additional simulation results

3.8 Summary

This chapter analyzed the use of orthonormal cipher matrix in encrypting images. It can be used to extract the relevant information, similar to Walsh Transform, Hadamard Transform, Discrete Cosine Transform, Discrete Sine Transform, Discrete Fourier Transform etc. In order to make the encryption more secure, the concept of private key has been introduced. Five methods of generating higher order orthonormal matrices have been suggested. Simulation results show the efficacy of the proposed algorithm.

Chapter 4

Matrix Exponentiation and Image Encryption

Chapter 4

Matrix Exponentiation and Image encryption

Previously, there has been an extensive study on various operations and their properties related to matrices. Some of these operations are addition, subtraction, scalar multiplication, multiplication and inversion. In this chapter, a new operation on matrices called **exponentiation** has been proposed.

Exponentiation of a matrix by another matrix is defined in the following section. It is a binary operation and is denoted by the operator '$**$'. In this chapter, all the properties of this operation has been presented. Moreover, it has been also used for encryption of images by asymmetric key encryption technique.

4.1 Exponentiation Operation over Z_p

4.1.1 Definition of exponentiation operator

Definition 4.1

Let A be the $l \times m$ and B be $m \times n$ matrices defined over Z_p.

$A**B$ is defined as

$$C = A**B = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{l1} & a_{l2} & \cdot & \cdot & \cdot & a_{lm} \end{bmatrix} ** \begin{bmatrix} b_{11} & b_{12} & \cdot & \cdot & \cdot & b_{1n} \\ b_{21} & b_{22} & \cdot & \cdot & \cdot & b_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{m1} & b_{m2} & \cdot & \cdot & \cdot & b_{mn} \end{bmatrix} \tag{4.1}$$

$$= \begin{bmatrix} c_{11} & c_{12} & \cdot & \cdot & \cdot & c_{1n} \\ c_{21} & c_{22} & \cdot & \cdot & \cdot & c_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{l1} & c_{l2} & \cdot & \cdot & \cdot & c_{ln} \end{bmatrix}$$

$$\text{where } c_{ij} = \prod_{k=1}^{m} a_{ik} **b_{kj} \text{ for } i = 1 \cdots l;\ j = 1, \cdots n \text{ and } c_{ij} \in Z_p \tag{4.2}$$

In this chapter, $**$ is used as normal exponentiation operator and $*$ is used as normal multiplication operator.

4.1.2 Proposed theorem on exponentiation identity matrix

Theorem 4.1

If A is a n × n square matrix, then there exists an exponentiation identity matrix I_e which is equal to the multiplicative identity matrix I.

$$\text{i.e. } A**I_e = A**I = A$$

Proof:

As per definition of exponentiation operator $A**B = C$ where

$$c_{ij} = \prod_{k=1}^{m} a_{ik} **b_{kj} = (a_{ij} **1) * \prod_{k=1,k\neq j}^{n} a_{ik} **0 = a_{ij}$$

Therefore $I_e = I$

4.1.3 Proposed theorem on exponentiation inverse of a matrix

Theorem 4.2

If A is a $n_1 \times n_2$, B is a $n_2 \times n_3$ and C is a $n_3 \times n_4$ matrix defined over Z_p

$$\text{then } (A**B)**C = A**(B*C)$$

Proof:

$$\text{Let } D = A**B \text{ and } E = D**C$$

$$D = \prod_{k=1}^{n_2} a_{ik} **b_{kl} \text{ for } i = 1, \cdots, n_1$$

$$E = \prod_{l=1}^{n_3} (\prod_{k=1}^{n_2} a_{ik} **b_{kl}) **c_{lj} \text{ for } i = 1, \cdots, n_1 \text{ and } j = 1, \cdots, n_4$$

$$= \prod_{k=1}^{n_2} a_{ik} ** \sum_{l=1}^{n_3} b_{kl} * c_{lj}$$

$$= A**(B*C)$$

Corollary

If A is a $n_1 \times n_2$, B is a $n_2 \times n_3$ and C is a $n_3 \times n_4$ matrix over Z_p then

$$(A **B) **C = A **(B * C) = A$$

if and only if C is the multiplicative inverse of B modulo $(p-1)$.

Proof:

By Theorem 4.2, $(A **B) ** C = A **(B * C)$ and since $a^{p-1} = 1$, $B * C = I$ only when $b_{ij} = 0$ mod (p-1) and $b_{ii} = 1$ mod$(p-1)$.

Therefore B $*$ C = I mod $(p-1)$.

N.B. Above theorem is not valid for p, if p is not a prime number.

4.2 Properties

The different properties which the exponentiation operator $**$ satisfies are mentioned below.

1. Closure : If A and B are two matrices defined over Z_p, then $A **B$ will also be defined over Z_p.

2. Associativity : The $**$ operator does not satisfy the associativity property as

$$(A **B) **C \neq A **(B **C)$$

This is evident from Theorem 4.2 .

3. Commutativity: The $**$ operator does not satisfy the commutativity property as

$$A **B \neq B **A$$

This is obvious by definition.

4. Existence of identity element: As per Theorem 4.1, exponentiation identity matrix exists and is same as multiplicative identity matrix.

5. Existence of inverse: Exponentiation inverse of a matrix is same as multiplicative inverse of a matrix which is evident from Theorem 4.2. Therefore, exponentiation inverse of a matrix can only exist if the matrix is non-singular.

4.3 Proposed algorithm for encryption and decryption

All the operations mentioned below are carried out in Z_p .

(a) Encryption

Step 1. Generate a self-invertible matrix 'A' in mod $(p-1)$ by any one of the methods mentioned in section 2.2 and section 2.3.

Step 2. Select 'n' degree polynomials with non-zero roots arbitrarily.

Step 3. Generate a matrix 'B' with eigen factors derived from Step 2 and using the method discussed in section 2.3.

Step 4. Determine the key matrix 'C' for encryption by the relation $C = A*B*A$.

Step 5. Divide the image into 8×8 blocks.

Step 6. Encrypt each block by the key matrix using exponentiation operation.

Step 7. Combine these blocks to form the encrypted image.

(a) Decryption

Step 1. Divide the encrypted image into 8×8 blocks.

Step 2. Generate decryption matrix 'D' by the relation $D = A * B^{-1} * A$ as discussed in section 2.3.

Step 3. Decrypt each block by exponentiation operation using the decryption

matrix.

Step 4. Form the decrypted image by combining these blocks.

4.4 Cryptanalysis

The proposed algorithm of encryption is similar to the algorithm discussed in chapter 2. Hence, the cryptanalysis done in chapter 2 holds good for this algorithm. Moreover, since encryption involves a very complex operation i.e. exponentiation, it is difficult to find the key matrix by plain-text, cipher-text or brute-force attack.

4.5 Results

The proposed algorithm for encryption and decryption was validated using simulation technique. In the simulation studies, image of 256 × 256 pixels with 8 bit encoding for each pixel is considered.

4.5.1 Simulation result of encryption in GF(p) ($p = 251$)

Let A be a self-invertible matrix selected randomly in GF(p).

$$A = \begin{bmatrix} 134 & 103 & 195 & 195 & 29 & 91 & 43 & 47 \\ 86 & 241 & 148 & 216 & 78 & 123 & 113 & 2 \\ 31 & 239 & 92 & 101 & 19 & 7 & 105 & 215 \\ 31 & 5 & 23 & 240 & 231 & 191 & 203 & 45 \\ 167 & 85 & 211 & 47 & 218 & 3 & 17 & 83 \\ 228 & 100 & 174 & 88 & 4 & 20 & 207 & 164 \\ 186 & 190 & 238 & 236 & 14 & 75 & 106 & 92 \\ 64 & 230 & 112 & 114 & 86 & 35 & 83 & 197 \end{bmatrix}$$

The private key matrix B (in GF(p)) has been generated by taking eigen factors $x-31$, $x-7$, $x-191$, x^3+x^2+1, $x-47$ and $x-127$ as defined in section 2.3. So, the private key B can be presented as

$$B = \begin{bmatrix} 31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 191 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 47 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 127 \end{bmatrix}$$

The public key matrix C (in GF(p)) is generated by using the relation $C = A * B * A$.

$$C = \begin{bmatrix} 220 & 83 & 167 & 10 & 47 & 191 & 233 & 1 \\ 83 & 192 & 19 & 88 & 31 & 157 & 208 & 161 \\ 165 & 129 & 196 & 214 & 197 & 89 & 83 & 237 \\ 240 & 180 & 240 & 2 & 63 & 126 & 138 & 18 \\ 210 & 100 & 220 & 67 & 42 & 210 & 73 & 212 \\ 162 & 170 & 156 & 98 & 184 & 47 & 242 & 190 \\ 113 & 195 & 249 & 70 & 141 & 179 & 209 & 113 \\ 47 & 155 & 131 & 200 & 89 & 161 & 228 & 244 \end{bmatrix}$$

The inverse of public key D (in GF(p)) is generated by using the relation $D = A * B^{-1} * A$

$$D = \begin{bmatrix} 127 & 6 & 108 & 102 & 176 & 48 & 97 & 28 \\ 25 & 228 & 225 & 143 & 18 & 163 & 66 & 21 \\ 46 & 218 & 89 & 138 & 78 & 246 & 215 & 132 \\ 91 & 85 & 133 & 61 & 67 & 131 & 245 & 29 \\ 210 & 100 & 190 & 9 & 92 & 136 & 244 & 28 \\ 28 & 230 & 214 & 92 & 120 & 185 & 104 & 4 \\ 163 & 235 & 209 & 15 & 138 & 41 & 181 & 175 \\ 197 & 115 & 121 & 45 & 192 & 239 & 72 & 158 \end{bmatrix}$$

256 × 256 cameraman image and iris flower image shown in Figures 2.1 and 4.1 are considered to carry out the simulation. Their histograms are presented in Figures 2.2 and 4.2. Since encryption is done in GF(p) field with $p = 251$, pixels of original image beyond 250 are rounded to 250. The images are rounded to 250 pixels and their histograms are shown in Figures 2.3, 2.4, 4.3 and 4.4 respectively.

Figure 4.1: Original iris flower image used for encryption

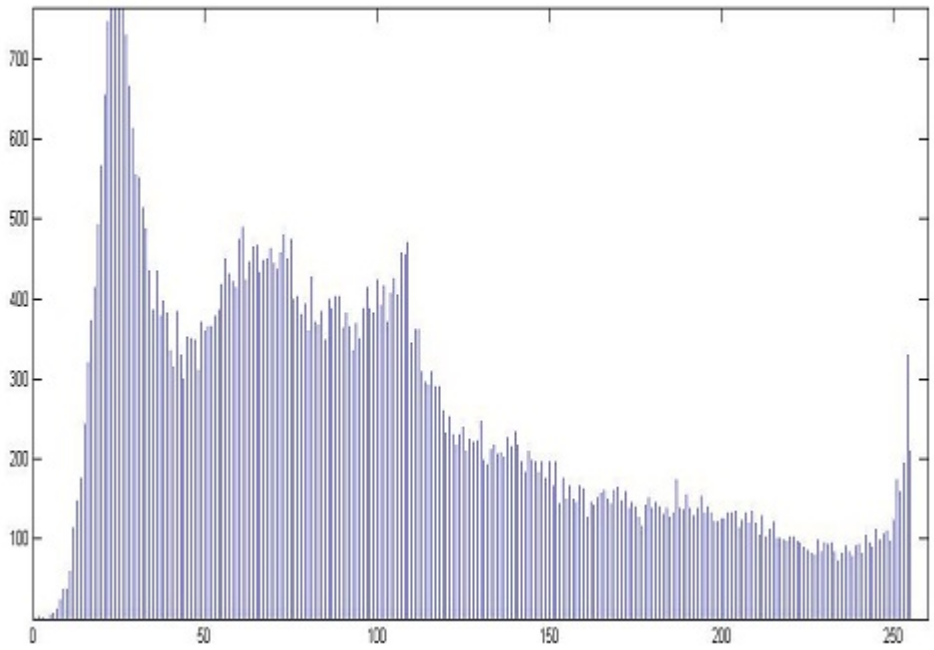

Figure 4.2: Histogram of original iris flower image

Figure 4.3: Iris flower image with pixels ≤ 250

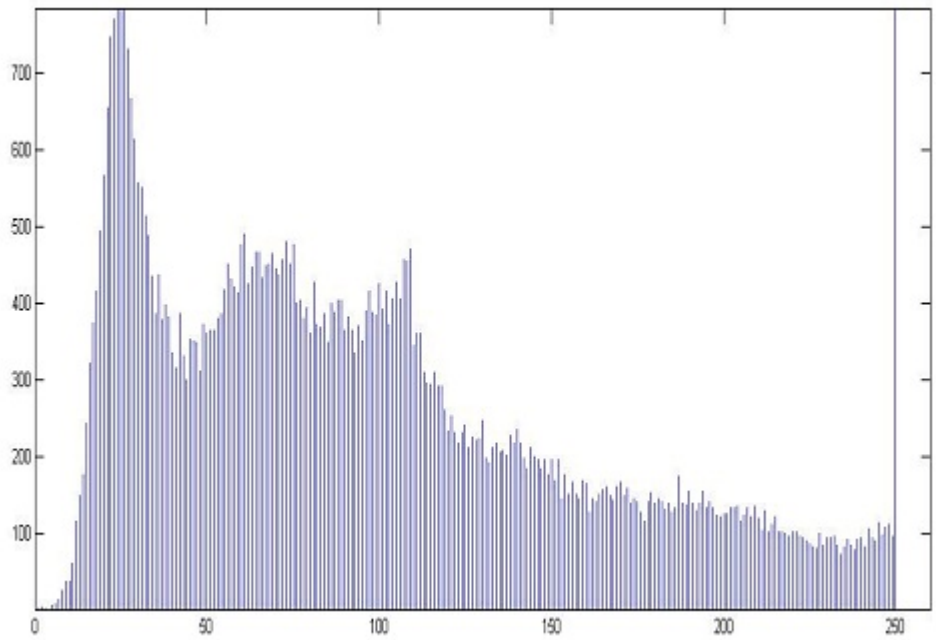

Figure 4.4: Histogram of iris flower image with pixels ≤ 250

The encryption algorithm proposed in section 4.3 is used to encrypt the cameraman image and iris flower image which are rounded to 250 pixels. The encrypted images and their corresponding histograms are shown in Figures 4.5, 4.6 and 4.7 respectively. On comparing the encrypted images with the original images, it can be seen that the encrypted images and their histograms do not bear any resemblance with the original images and their histograms.

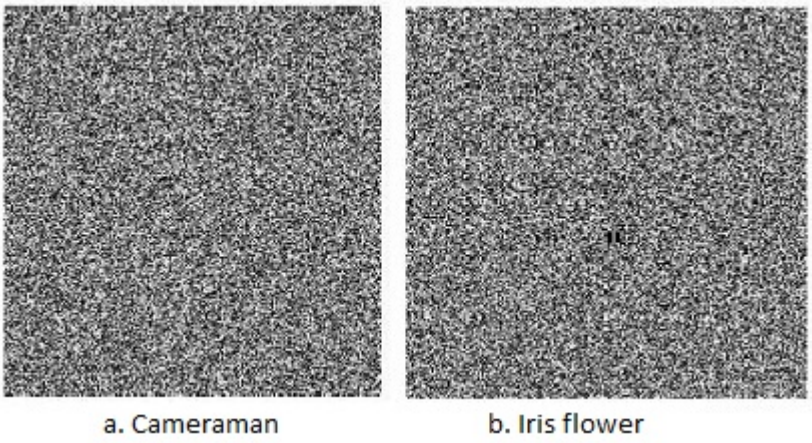

Figure 4.5: Encrypted image of cameraman and iris flower using encryption algorithm in section 4.3

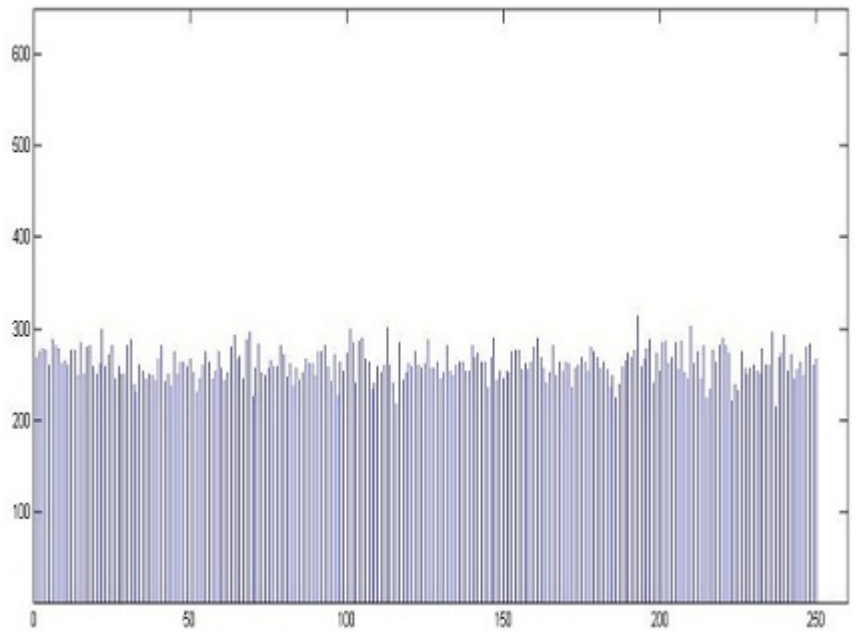

Figure 4.6: Histogram of encrypted image of cameraman
using encryption algorithm in section 4.3

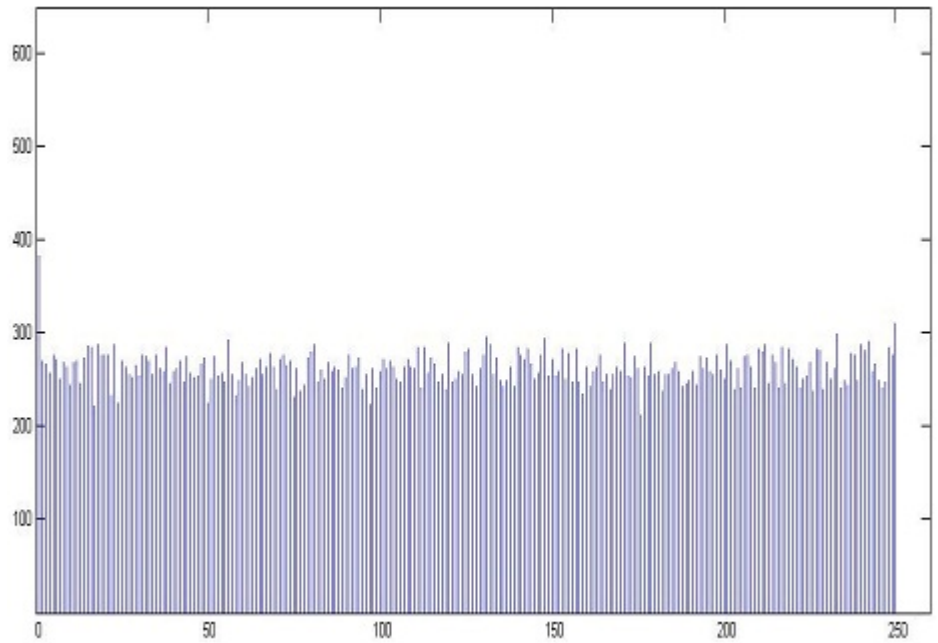

Figure 4.7: Histogram of encrypted image of iris flower
using encryption algorithm in section 4.3

The encrypted images are decrypted by using the decryption algorithm presented in section 4.3. The decrypted images and their corresponding histograms are shown in Figure 4.8, 4.9 and 4.10 respectively which are same as the original image rounded to 250 pixels and its histogram.

Figure 4.8: Decrypted images of cameraman and iris flower using decryption algorithm in section 4.3

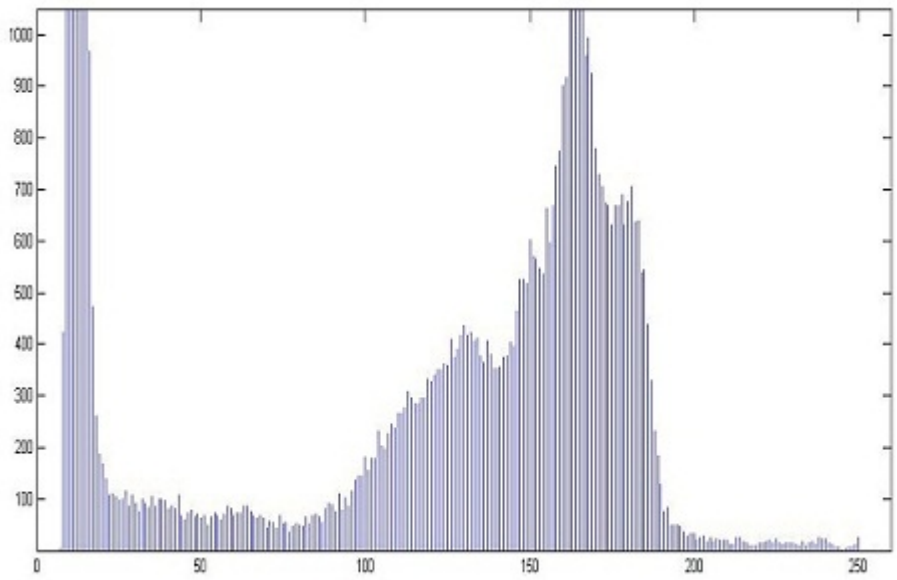

Figure 4.9: Histogram of decrypted image of cameraman using decryption algorithm in section 4.3

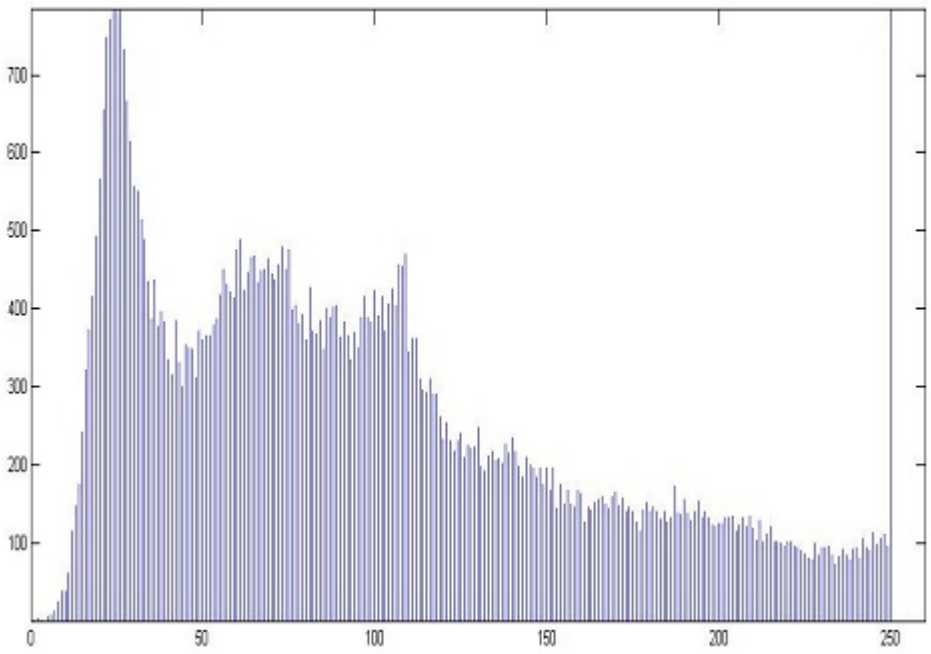

Figure 4.10: Histogram of decrypted image of iris flower
using decryption algorithm in section 4.3

The proposed algorithm is also tested with other images. Some of the simulation results are shown in Figure 4.11, where Figure 4.11(a,b,c) represent the original images of Baboon, Eveface and Barbara. The corresponding histograms of the original images are shown in Figure 4.11(d,e,f). The encryption algorithm proposed in section 4.3 is used for encrypting the images. The encrypted images are shown in Figure 4.11(g,h,i) and their histograms are represented in Figure 4.11(j,k,l).

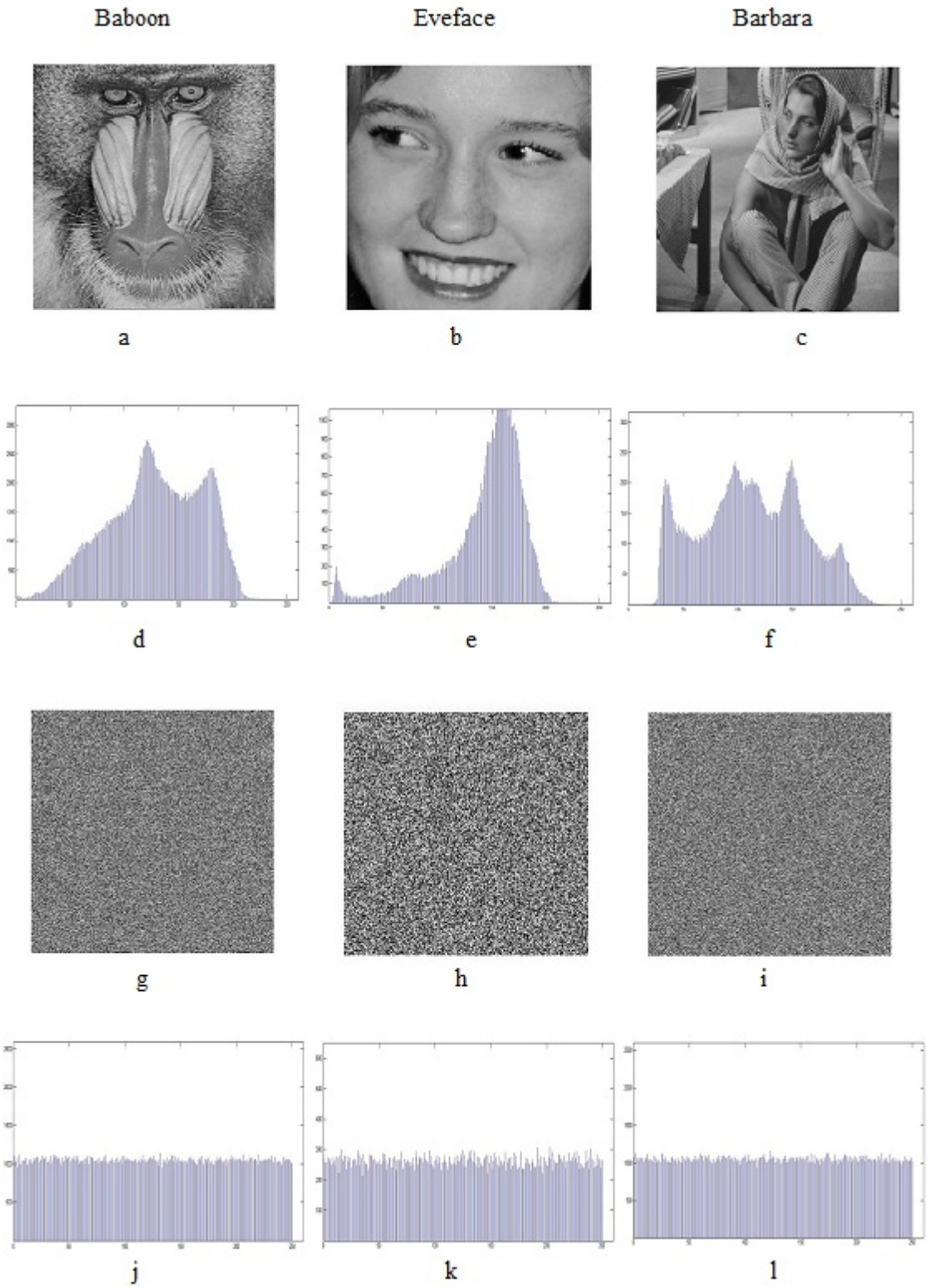

Figure 4.11: Additional simulation results

4.6 Summary

In this chapter, a new operation i.e. exponentiation of square matrices whose elements are in GF(p) (where $p = 251$) has been defined and two theorems are also proposed which will be helpful for image encryption. This operation has been seen to be effective in encryption of images.

Chapter 5

Image Encryption

by

Integer Chaotic Sequence

Chapter 5

Image Encryption by Integer Chaotic Sequence

In chapter 4, a new operation called exponentiation operation was introduced to make image encryption more robust and the private keys used in this chapter(and previous chapters) can be found out by standard algorithms used to find the inverse of matrices. But, it takes considerable amount of time to find the private keys which makes the image encryption technique robust. In order to enhance the robustness of the image encryption technique, the image is encrypted in two stages in this chapter. In the first stage, the image is encrypted using a novel integer chaotic sequence algorithm and then second stage encryption is done using one of the encryption algorithms mentioned in chapters 2,3 and 4.

A chaotic sequence is a non-periodic, non-converging random sequence. Its initial value can be varied to produce different sequences. Since, the chaotic sequence is deterministic and reproducible, it can be used conveniently for encryption of any data sequence.

5.1 Encryption by chaotic sequence

A Non-linear chaotic algorithm uses power function and tangent function instead of a linear function. These functions produce highly uncorrelated random sequence. This algorithm requires a set of parameters which are generated by using a one-time-one-password. Moreover, this algorithm has the advantage of a large key space and an improved security [25].

Image can be encrypted using a combination of different permutation techniques. The main idea is that an image can be viewed as an arrangement of bits, pixels and blocks. The intelligible information present in an image is due to the correlations among the bits, pixels and blocks in a particular arrangement. Security of image encryption can be improved by decreasing the correlation among the bits, pixels and blocks using certain permutation techniques. It is observed that the permutation of bits is effective in reducing the correlation thereby decreasing the perceptual information, whereas the permutation of pixels and blocks are good at producing higher level security compared to bit permutation [52].

A large external key is used to generate two chaotic logistic maps. It can be used to generate an efficient chaos based stream cipher for image encryption. In this algorithm, the parameters of second stage encryption depend on the previous encrypted data. For mixing the current encryption parameters with previously encrypted data, an encryption algorithm which uses an iterative cipher module based feedback and data-dependent input mechanism is used. The secret key is then changed after encryption of each pixel of the image, which makes the cipher more robust against any attack. This is an efficient encryption algorithm for real time [26].

In the traditional method, binary sequences are used for image permutation. Chaos sequences are the real valued sequences. In order to convert a chaos se-

quence into binary sequence an extra logic operation is required which generates the integer pseudo random number for image permutation.

Blowfish encryption algorithm is used to encrypt an image and the resulting image is further encrypted by an image encryption approach that uses block-based encryption. A seed is used as a key to generate a pseudo random sequence. The image is then divided into number of blocks and using this pseudo random sequence, the pixel position of the image is transformed. Using this technique the correlation among the pixels decreases and entropy increases [48].

Another technique of image encryption uses a combination of permutation technique followed by normal encryption method. This permutation technique is based on the combination of image permutation and a popular encryption algorithm called RijnDael. The image is divided into number of blocks and these blocks are then rearranged into permuted image. The resulting image is encrypted using the RijnDael algorithm. The correlation between image elements decreases significantly by using the combination technique and hence higher entropy is achieved [52].

The Shuffle Encryption Algorithm (SEA) applies nonlinear s-box byte substitution. Following this, a shuffling operation is performed which partially depends on the input data and the given key. The statistical analysis is done using histograms, correlation and covariance with which security of encryption algorithm is analyzed [36].

Image encryption can be done by selecting specific higher frequencies of DCT coefficients which are the characteristic values and then these DCT coefficients are encrypted. These encrypted blocks are shuffled using a pseudo-random bit sequence. This method reduces the computational requirements for huge volumes of images. In this method first the image is decomposed into 8×8 blocks. Then

these blocks are transformed from the spatial domain to frequency domain by the DCT and the DCT coefficients of higher frequencies are encrypted using Non-Linear Shift Back Register (stream cipher) [11]. The concept behind the technique is that, the detailed information of an image are embedded in the higher frequencies, which the human eye cannot perceive. The algorithm is lossless and this technique reduces time complexity.

Enhanced Image encryption algorithm by the chaos sequence is used to permute the blocks of the image. Chaotic systems are sensitive to the initial condition. Minor variation of initial condition produces a different sequence. The pixels of the image are permuted on the basis of index position of the chaotic sequence and then the blocks of image are permuted using this chaotic sequence by mapping it with index position and encryption is done. Slight variation of the initial condition will lead to a different encrypted image and thereby making the encryption algorithm more robust [38].

5.2 An overview of chaotic sequence

5.2.1 Chaos Theory

Chaos theory is now being applied to various fields of engineering, basic science like physics, philosophy related to field of mathematics and also economics. Here the focus is on the behavior of dynamic systems that are highly sensitive to initial conditions commonly referred to as the butterfly effect. Small variation in the initial conditions yields diverging results. The systems are deterministic without involvement of random elements. The deterministic nature makes the system predictable. Chaotic behavior is also evident in natural systems like weather. In common usage, “chaos” means “a state of disorder”.

5.2.2 Chaotic System

Chaotic system is a complex system and is highly dependent on initial conditions. For a system to be deterministic and accurate, the initial conditions must be known and to a certain level of accuracy. Following are various methods used for mapping the chaotic system.

Logistic map

- $X_{n+1} = AX_n(1 - X_n)$
- Usual parameter: $A = 4$

Hĕnon map

- $X_{n+1} = 1 + Y_n - aX_n^2$
- $Y_{n+1} = bX_n$
- Usual parameters: $a = 1.4, b = 0.3$

Chirikov (standard) map

- $X_{n+1} = X_n + Y_{n+1} mod 2p_i$
- $Y_n + 1 = Y_n + ksin \times nmod 2p_i$
- Usual parameter: $k = 1$

Lorenz attractor

- $\frac{dx}{dt} = \sigma(y - x)$
- $\frac{dy}{dt} = -xz + rx - y$
- $\frac{dz}{dt} = xy - bz$
- Usual parameters: $\sigma = 10, r = 28, b = \frac{8}{3}$

Rössler attractor

- $\frac{dx}{dt} = -y - z$
- $\frac{dy}{dt} = x + ay$
- $\frac{dz}{dt} = b + z(x - c)$
- Usual parameters: $a = b = 0.2, c = 5.7$

Ueda attractor

- $\frac{dx}{dt} = y$
- $\frac{dy}{dt} = -x^3 - ky + Bsin(z)$
- $\frac{dz}{dt} = 1$
- Usual parameters: $B = 7.5, k = 0.05$

Simplest quadratic dissipative chaotic flow

- $\frac{dx}{dt} = y$
- $\frac{dy}{dt} = z$
- $\frac{dz}{dt} = -Az + y_2x$
- Usual parameter: $A = 2.017$

Simplest piecewise linear dissipative chaotic flow

- $\frac{dx}{dt} = y$
- $\frac{dy}{dt} = z$
- $\frac{dz}{dt} = -Az - y - |x| + 1$
- Usual parameter: $A = 0.6$

5.2.3 Chaotic Sequence

A chaotic sequence is non-converging, non-periodic sequence and exhibits noise-like behavior. The initial value can be varied to produce number of uncorrelated, random-like, yet deterministic and reproducible signal sequences. These sequences called the chaotic sequences are real valued sequences which can be converted into integer valued sequences which are effective for pixel permutation during image encryption.

One of the simplest and most widely studied nonlinear dynamic systems capable of exhibiting chaos is the logistic map.

Properties of chaotic sequence

For any dynamic function f mapping V $\rightarrow$ V to be classified as chaotic, it must have the following properties-

- periodic points are dense in V
- f is topologically transitive
- f has sensitive dependence on initial conditions

Consider our function $f(x) = \mu \times x(1-x)$ where μ is the control parameter. This function maps [0,1] to [0,1] and matches all above criteria.

Dense periodic points A set is called dense when there is another number between any two numbers. For instance, between 2 and 2.1 there is 2.05. But, between 2 and 2.05 there is 2.005. Therefore, that a dense set has infinite number of points.

Topological transitivity When we pick up any open interval of any size within [0,1], there will exist a point in the selected interval that will jump to another interval within [0,1] under iteration.

Sensitive dependence on initial conditions Different initial conditions produce different chaotic sequences. The sensitivity dependence implies that no two chaotic sequences will converge in a periodic cycle of iteration of any length. However, the chaotic sequences may converge in a non-periodic cycle of iteration.

Logistic map The logistic map is a polynomial mapping of degree 2, which is a simple example of chaotic behavior which is obtained in a very simple non-linear dynamical equation. Mathematically, the logistic map is written as,

$$x_{k+1} = x_k \times \mu \times (1 - x_k) \text{ where, } 1 < \mu < 4 \text{ and } x_k \in [0, 1]$$

Analysis of the variation of sequence for different values of μ By varying the parameter μ, the following behavior is observed.

• With μ between 0 and 1, the sequence will die, independent of the initial value.

• With μ between 1 and 2, the sequence will converge to $\mu - \frac{1}{\mu}$, independent of the initial value.

• With μ between 2 and 3, the sequence will converge to $\mu - \frac{1}{\mu}$, but it will fluctuate around that value for some time. The rate of convergence is linear, except for μ =3. The convergence will be slow if μ=3.

• With μ between 3 and 1+ $\sqrt{6}$ (approximately 3.45), for almost all initial conditions the sequence will approach permanent oscillations between two values. These two values are dependent on μ.

• With μ between 3.45 and 3.54 (approximately), for all initial conditions the sequence will approach permanent oscillations between four values. These four values are dependent on μ.

• With μ increasing beyond 3.54, for almost all initial conditions the sequence will approach oscillations 8 values, these eight values are dependent on μ.

• At μ approximately 3.57 is the onset of chaos, for almost all initial conditions

there cannot be any oscillations of finite period. Slight variations in the initial population yield dramatically different results over time, a prime characteristic of chaos.

• Most values beyond 3.57 exhibit chaotic behavior, but there are still certain isolated ranges of μ that show non-chaotic behavior. Beyond $\mu = 4$, the values eventually leave the interval [0,1] and diverge for almost all initial values.

5.3 Proposed scheme

Two stages of encryption are introduced here. In the first stage, chaotic sequence is generated in GF(p). Each row of pixels of the gray image is multiplied by a separate chaotic sequence. The second stage of encryption is performed by multiplying the modified data of the first stage with a 8×8 cipher matrix generated by any one of the algorithm discussed in chapters 2, 3 or 4. Decryption is carried out by multiplying the final encrypted data with the multiplicative inverse of the cipher matrix and then the multiplicative inverse of the chaotic sequence is used for second stage of decryption.

5.3.1 Generation of Integer Chaotic Sequence

$$x(i) = x(i-1) * \mu * (1 - x(i-1))$$

$$z_i = (x(i) * Q)$$

$$z = round(z_i * N)$$

$$y(i) = mod(z, p)$$

where initial value of x i.e. $x(1) = 0.2$ and μ=3.8, Q=105, $N = 10^5$ and $p = 251$. Since different chaotic sequence can be generated by varying initial value $x(1)$[0,1],

μ [3.6,4) and Q[104,106] these can be considered as private keys. Block-based transformation could have been used but here direct approach is used.

5.3.2 Algorithm of Encryption and Decryption

(a) Encryption

Step 1. All the pixel values p_{ij} are multiplied by $y(256 * (i-1)+j)$ i.e. $p_{ij} = p_{ij} * y(256 * (i-1)+j) \bmod p$ to obtain cipher1 image pixel values cp_{ij}.

Step 2. Divide this transformed cipher1 image into 8 $\times$ 8 blocks.

Step 3. Select a public key matrix from one of the previous chapters(chapters 2, 3 or 4)and encrypt each block by the key matrix.

Step 4. Combine the blocks to get the cipher2 image.

(b) Decryption

Step 1. Divide the encrypted image into 8 $\times$ 8 blocks.

Step 2. Decrypt each block by using the decryption matrix corresponding to the public key matrix used in encryption algorithm.

Step 3. Form the decrypted image by combining these blocks. The image thus obtained in this step is the cipher1 image.

Step 4. Multiply cp_{ij} obtained in Step 1 with the multiplicative inverse of $y(256 * (i-1)+j)$ to obtain the original image.

5.4 Cryptanalysis

An algorithm based on 2-stage encryption is presented in this chapter. The second stage of encryption uses any one of the methods suggested in chapters 2,3 or 4. So, cryptanalysis of this stage of encryption is same as the cryptanalysis in chapter 2. Another level of security is introduced by the first stage of encryption. In this stage, the pixels of the image are multiplied with the integer

chaotic sequence. Generation of integer chaotic sequence requires 3 parameters ($x(1), \mu$ and Q) which act as private keys. It is very difficult to find out the private keys even if the chaotic sequence is known.

5.5 Results

The proposed algorithm for encryption and decryption is validated using simulation technique. In the simulation studies, image of 256×256 pixels with 8 bit encoding for each pixel is considered.

The following chaotic sequence y(i) (where $1 \leq i \leq 256$) is generated for x(1)=0.2, μ=3.8, Q=105, N=10^5 and p=251 and is used to transform the original image into cipher1 image.

171 215 12 15 10 159 83 220 88 80 229 49 96 197 205 245 139 1 125 34 81 144 62 136 171 116 47 34 126 26 30 196 67 151 200 169 100 146 138 128 154 154 211 104 22 7 30 101 242 83 193 186 94 130 204 148 112 19 230 123 51 108 119 4 244 235 134 173 102 91 230 178 195 228 1 145 210 115 224 94 53 152 58 63 95 237 245 226 166 247 49 203 208 210 151 124 92 176 19 37 204 5 216 26 147 45 105 19 123 224 173 54 37 176 32 211 104 98 104 188 1 248 57 139 142 144 3 92 124 160 232 159 69 215 226 177 90 168 123 172 201 33 191 167 131 82 106 144 229 30 200 27 172 33 87 217 165 141 29 199 231 152 62 120 43 35 180 40 198 126 57 160 66 90 241 50 68 135 144 54 79 232 181 201 32 191 167 132 82 107 142 233 27 209 33 154 214 136 235 110 53 194 222 229 34 118 156 34 181 59 233 37 192 1 190 137 157 104 72 138 213 26 19 40 132 82 96 34 224 170 176 145 137 185 87 184 226 190 213 195 108 183 215 249 112 164 63 132 141 14 125 159 174 127 83 94

Following is the public key matrix (C) in GF(p)(generated in chapter-2) which is used to transform the cipher1 image into cipher2 image.

$$C = \begin{bmatrix} 45 & 95 & 198 & 73 & 140 & 150 & 16 & 25 \\ 147 & 22 & 181 & 104 & 147 & 57 & 165 & 240 \\ 29 & 202 & 104 & 205 & 70 & 228 & 153 & 74 \\ 98 & 196 & 173 & 107 & 23 & 97 & 9 & 188 \\ 68 & 150 & 78 & 25 & 213 & 4 & 88 & 12 \\ 201 & 214 & 113 & 70 & 210 & 173 & 81 & 27 \\ 129 & 44 & 100 & 120 & 129 & 189 & 161 & 156 \\ 8 & 63 & 52 & 165 & 12 & 56 & 77 & 79 \end{bmatrix}$$

The inverse of public key C (in GF(p)) denoted by D is used to get back the cipher1 image from the cipher2 image.

$$D = \begin{bmatrix} 131 & 224 & 48 & 1 & 225 & 161 & 127 & 133 \\ 106 & 184 & 221 & 181 & 25 & 41 & 145 & 248 \\ 158 & 189 & 81 & 212 & 65 & 197 & 6 & 44 \\ 88 & 195 & 40 & 103 & 146 & 52 & 38 & 234 \\ 185 & 58 & 41 & 174 & 104 & 131 & 187 & 157 \\ 4 & 132 & 177 & 54 & 138 & 51 & 136 & 10 \\ 213 & 245 & 225 & 122 & 180 & 149 & 43 & 173 \\ 221 & 146 & 173 & 191 & 226 & 131 & 133 & 120 \end{bmatrix}$$

Mentioned below is Z(i), which is the multiplicative inverse of y(i)

where $1 \leq i \leq 256$ which is used to obtain the original image from cipher1 image.

160 244 21 67 226 30 124 170 174 91 57 41 34 79 60 209 186 11 249 96 31 190 166 24 160 132 235 96 2 29 159 73 15 128 187 101 123 98 231 151 207 207 69 70 194

36 159 169 223 124 238 139 243 56 16 212 65 185 239 100 64 86 135 63 215 47 133 74 32 80 239 55 121 120 1 206 202 227 158 243 90 180 13 4 37 233 209 10 62 188 41 183 35 202 128 83 221 87 185 95 16 201 43 29 181 106 153 185 100 158 74 172 95 87 102 69 70 146 70 247 1 167 229 186 175 190 84 221 83 171 66 30 211 244 10 78 53 127 100 54 5 213 46 248 23 150 45 190 57 159 187 93 54 237 176 155 143 162 26 111 138 180 166 228 216 208 152 182 161 2 229 171 232 53 25 246 48 119 190 172 197 66 147 5 102 46 248 116 150 61 175 237 93 245 213 207 156 24 47 89 90 22 225 57 96 117 214 96 147 234 237 95 17 1 144 11 8 70 129 231 33 29 185 182 116 150 34 96 158 220 87 206 11 19 176 236 10 144 33 121 86 203 244 125 65 75 4 116 162 18 249 30 88 168 124 243

256 $\times$ 256 cameraman image and crowd image shown in Figures 2.1 and 5.1 respectively are considered to carry out the simulation. Their histograms are presented in Figures 2.2 and 5.2 respectively. Since encryption is done in GF(p) with $p = 251$, pixels of original image beyond 250 are rounded to 250. The images are rounded to 250 pixels and their histograms are shown in Figures 2.3, 5.3, 2.4 and 5.4 respectively.

Figure 5.1: Original crowd image used for encryption

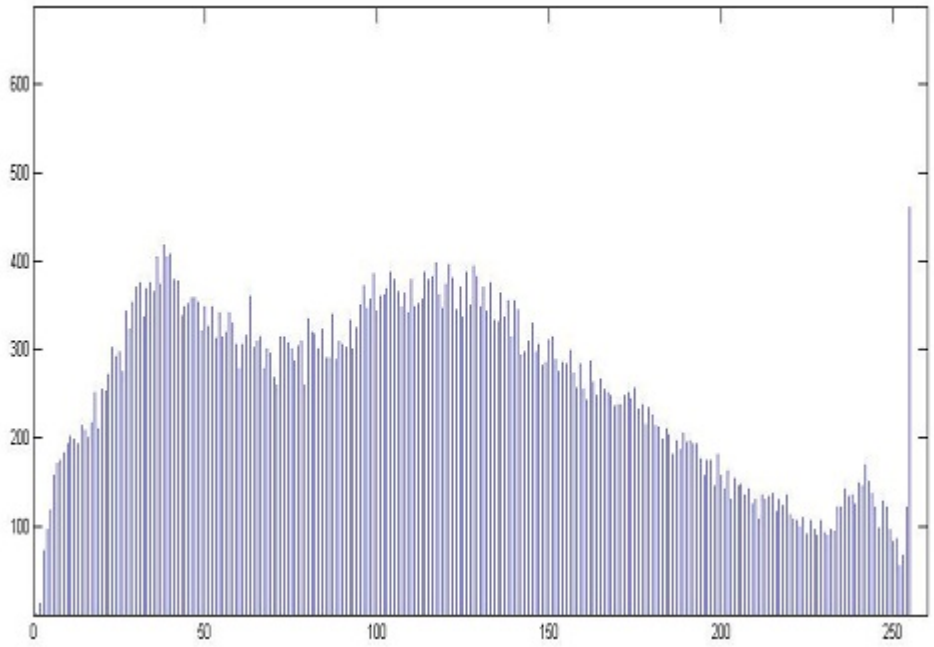

Figure 5.2: Histogram of original crowd image

Figure 5.3: Crowd image with pixels ≤ 250

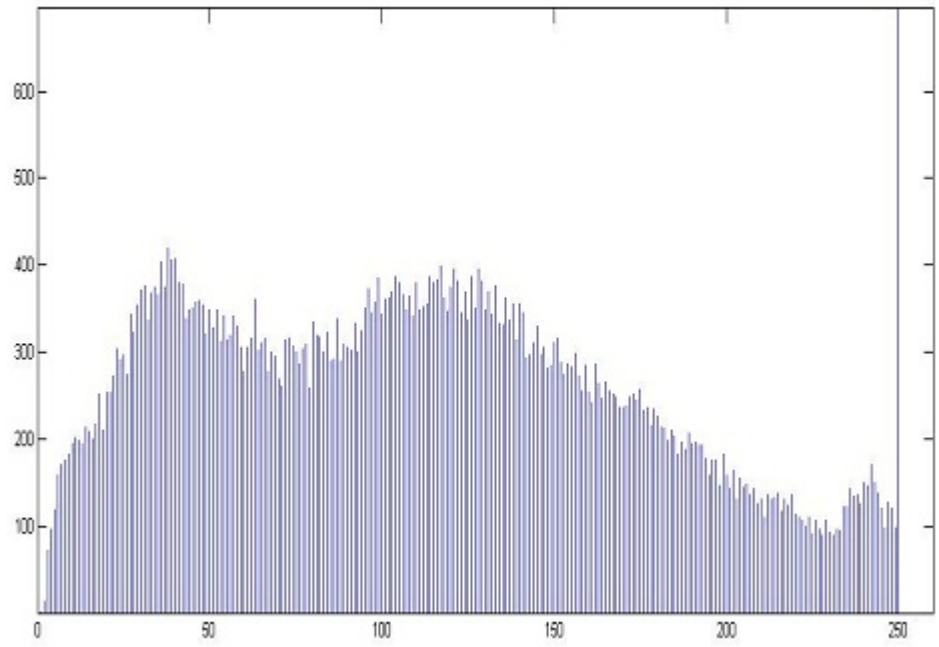

Figure 5.4: Histogram of crowd image with pixels ≤ 250

The encryption algorithm proposed in section 5.3.2 is used to encrypt the cameraman image and crowd image which are rounded to 250 pixels. The encryption algorithm consists of 2 stages. In the first stage, the image is encrypted using the chaotic sequence(y). The encrypted images and their corresponding histograms are shown in Figures 5.5, 5.6 and 5.7 respectively. On comparing the encrypted images with the original images it can be seen that the encrypted images and their histograms do not bear any resemblance with the original images and their histograms.

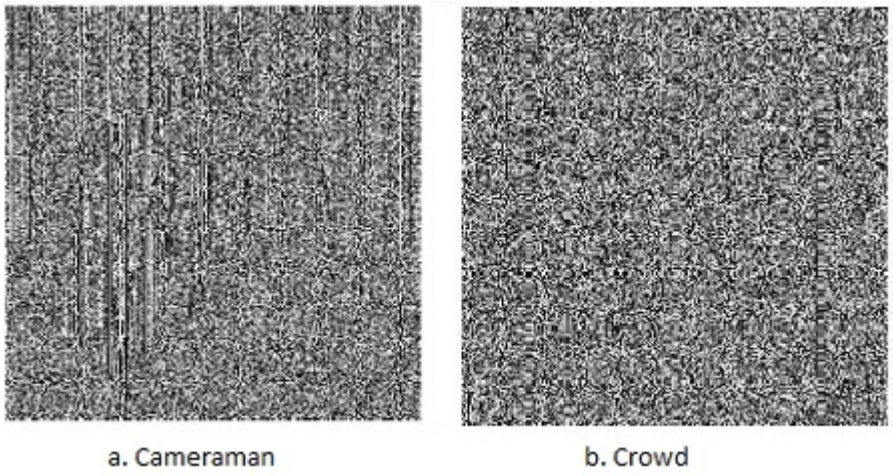

Figure 5.5: Encrypted image of cameraman and crowd using chaotic sequence y

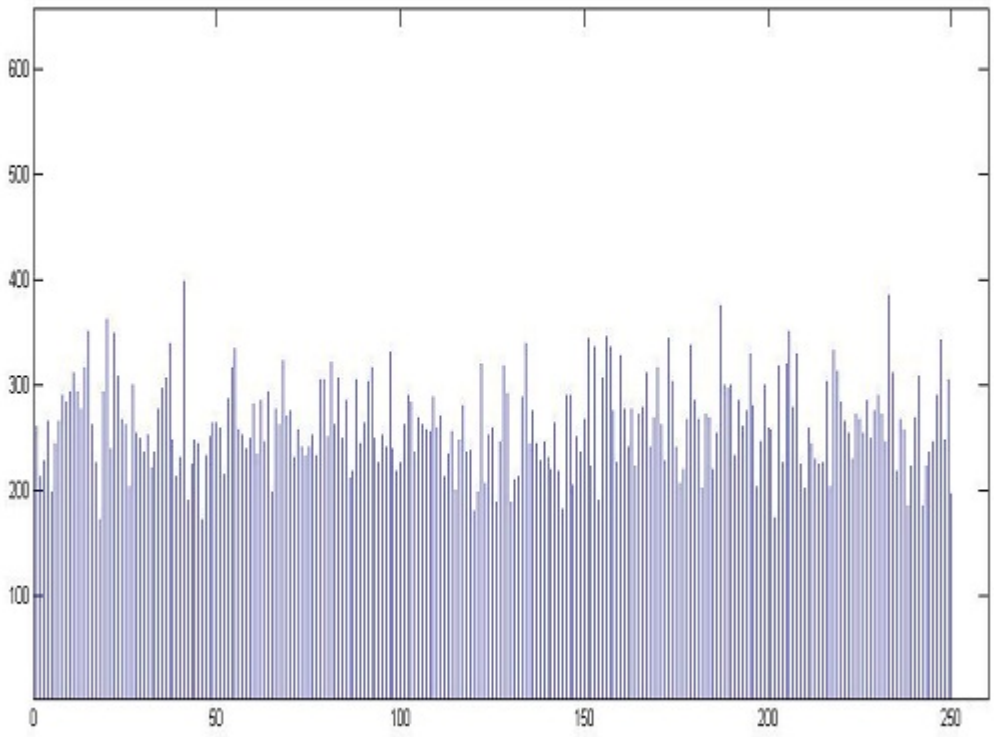

Figure 5.6: Histogram of encrypted image of cameraman using chaotic sequence y

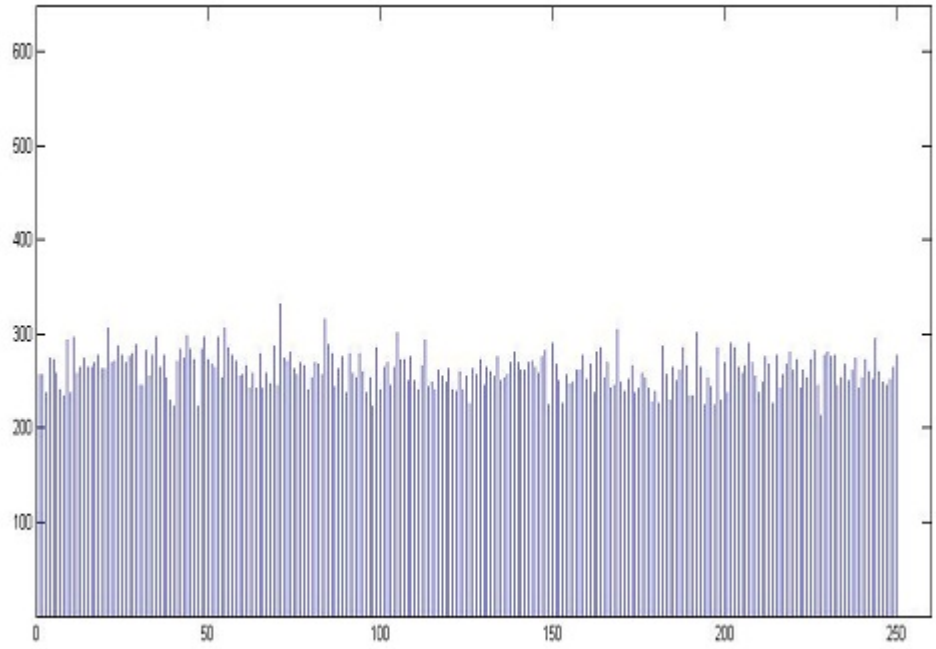

Figure 5.7: Histogram of encrypted image of crowd using chaotic sequence y

In the second stage of encryption, the resultant images of first stage is again encrypted using the public key C. The images after second stage encryption are shown in Figure 5.8 and their corresponding histograms are shown in Figures 5.9 and 5.10 respectively.

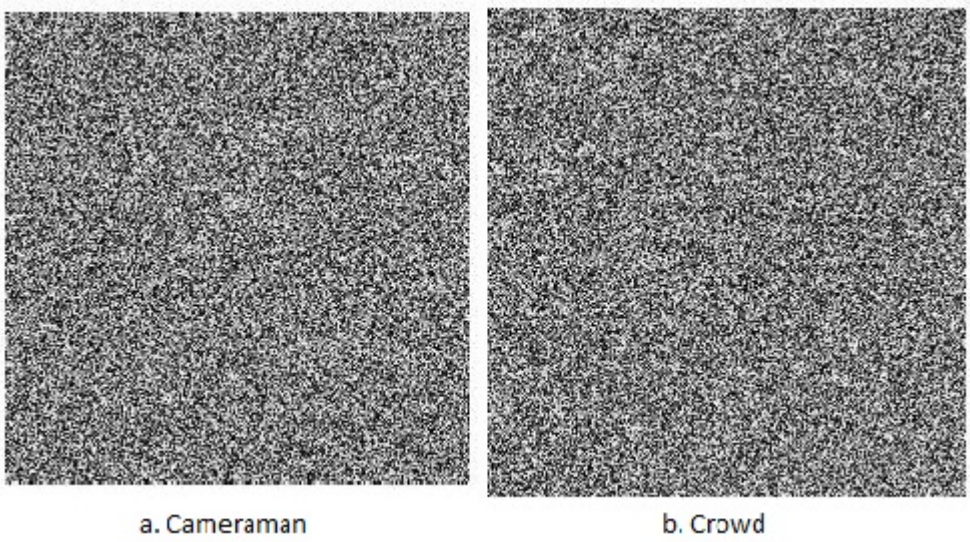

Figure 5.8: Images of cameraman and crowd after second stage encryption using public key C

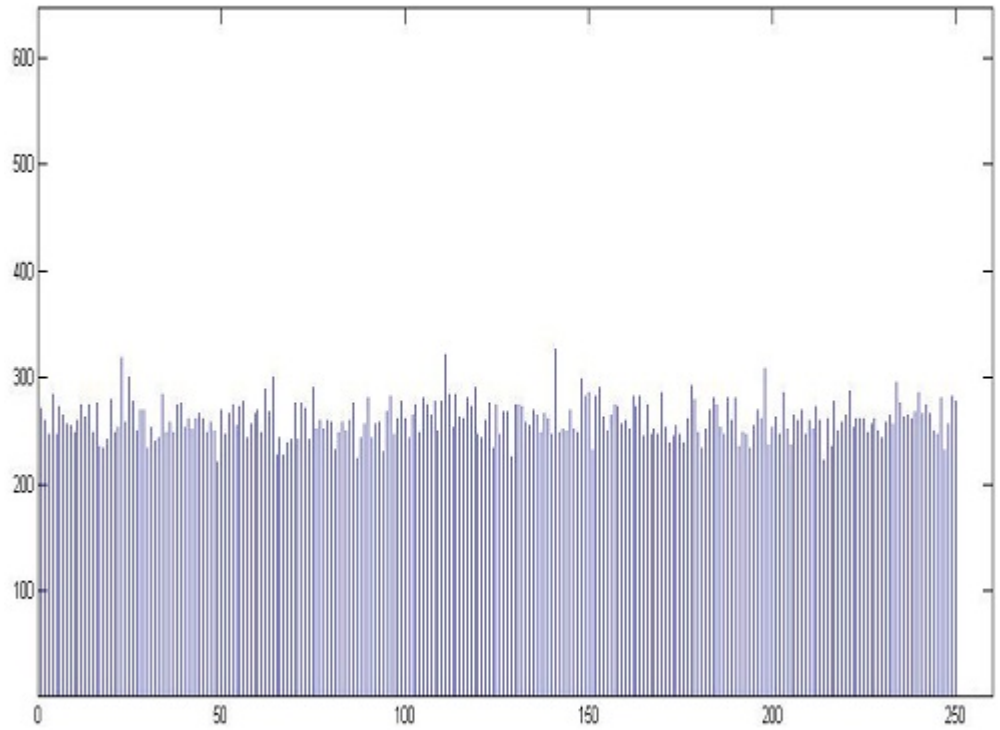

Figure 5.9: Histogram of image of cameraman after second stage encryption

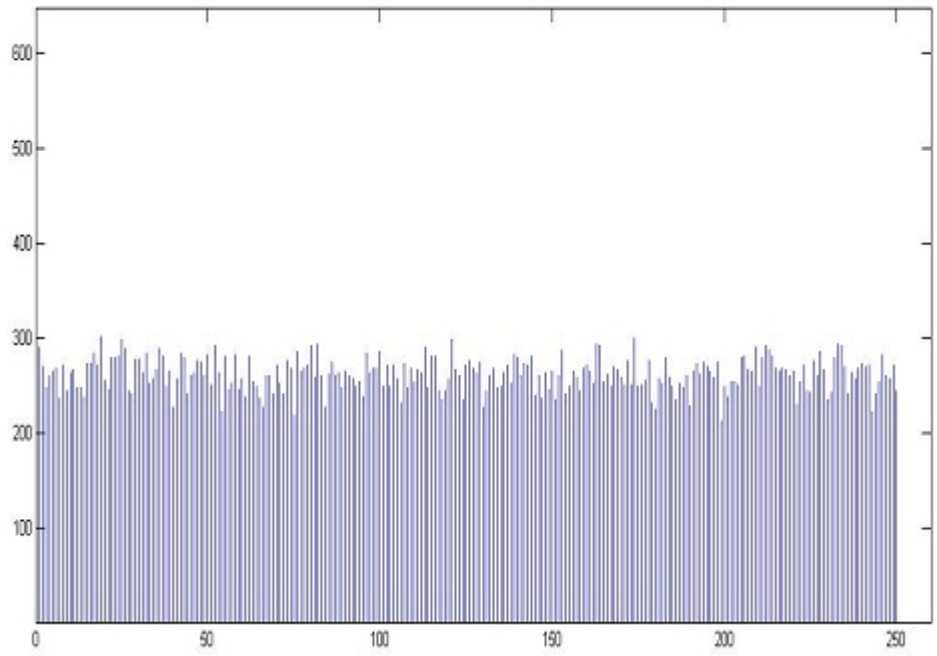

Figure 5.10: Histogram of image of crowd after second stage encryption

In the first stage decryption, the images are decrypted using the inverse of public key i.e. D. The images and their corresponding histograms are shown in Figures 5.11, 5.12 and 5.13 respectively.

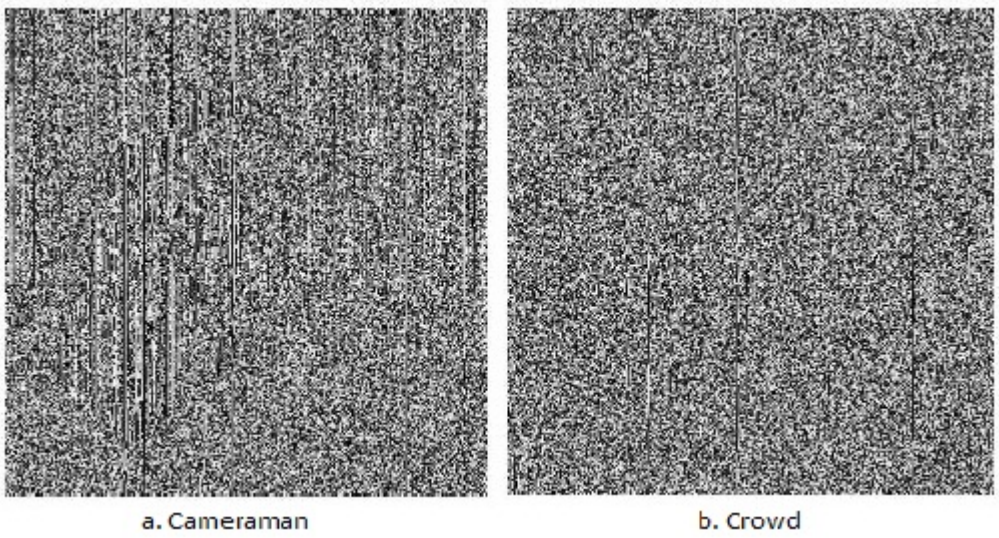

Figure 5.11: Image of cameraman and crowd after first stage decryption using private key D

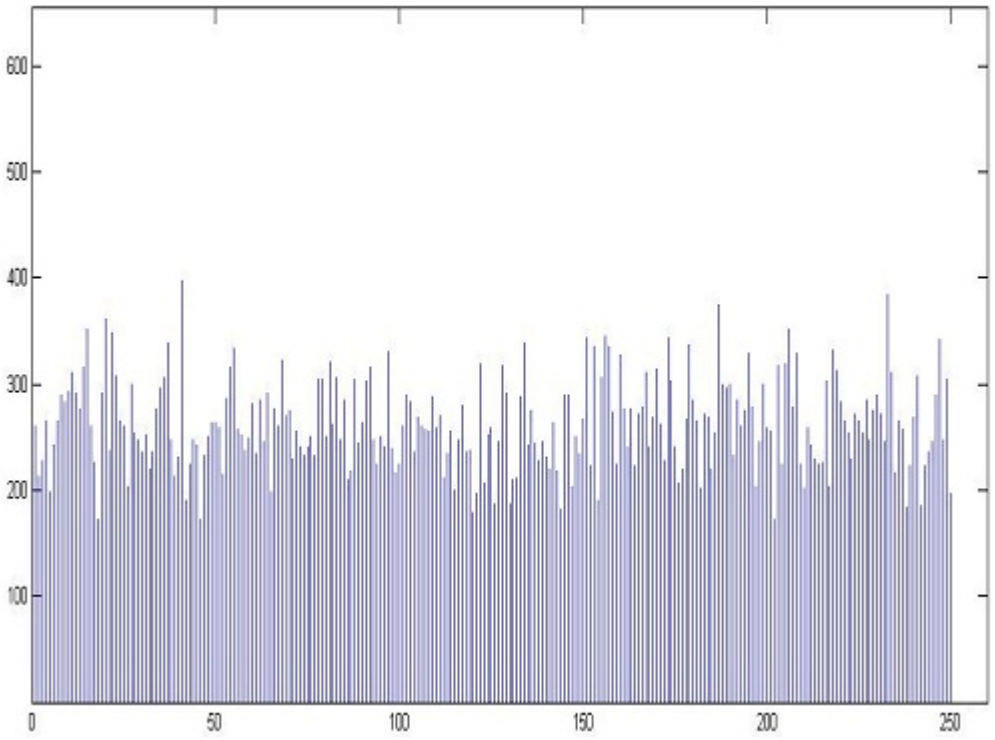

Figure 5.12: Histogram of image of cameraman after first stage decryption

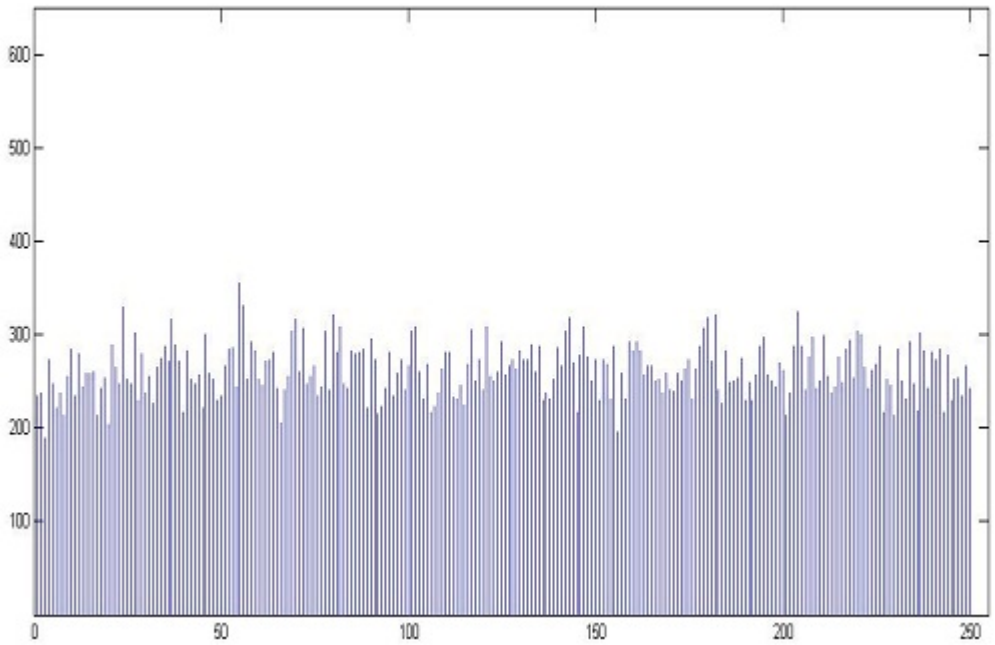

Figure 5.13: Histogram of image of crowd after first stage decryption

The original images are obtained after the second stage decryption using inverse of chaotic sequence i.e. Z. The images after second stage decryption and their corresponding histograms are shown in Figures 5.14, 5.15 and 5.16 respectively.

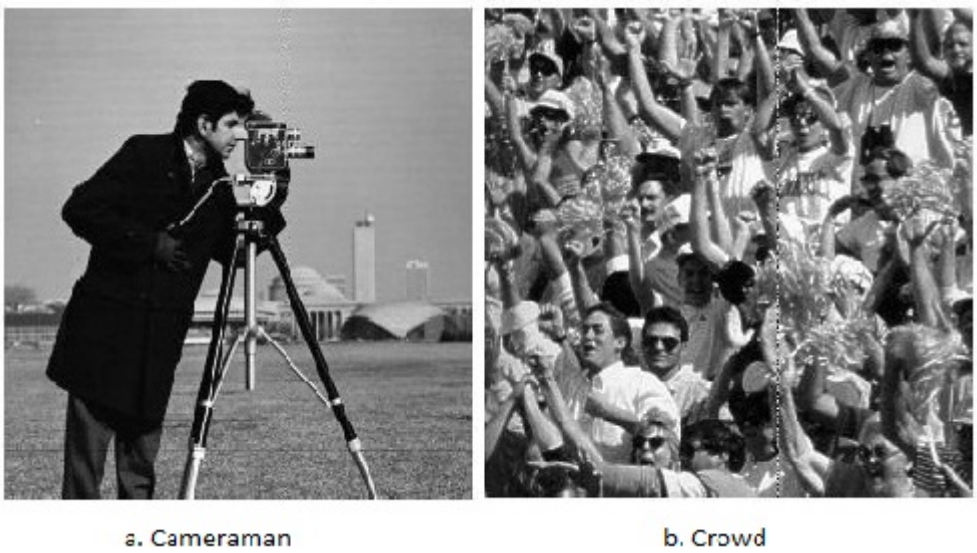

Figure 5.14:Image of cameraman and crowd after second stage decryption using Z

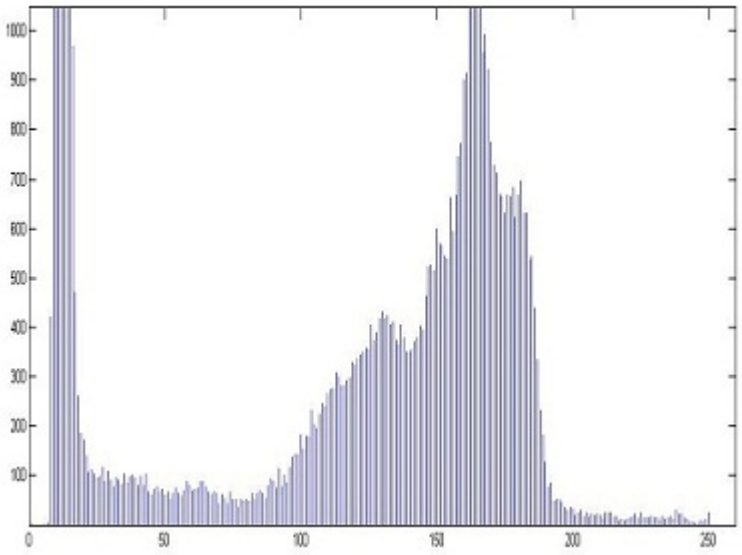

Figure 5.15: Histogram of image of cameraman after second stage decryption

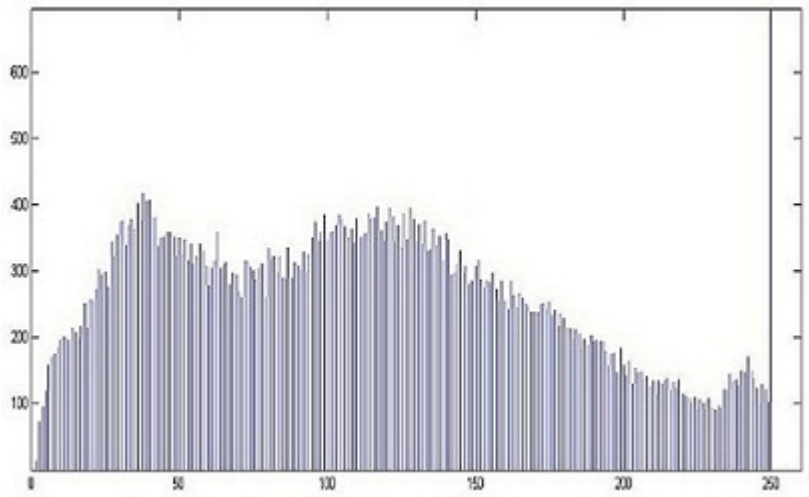

Figure 5.16: Histogram of image of crowd after second stage decryption

The proposed algorithm is also tested with other images. Some of the simulation results are shown in Figure 5.17, where Figure 5.17(a,b,c) represent the original images of Iris flower, Pepper and Lena. The corresponding histograms of the original images are shown in Figure 5.17(d,e,f). The encryption algorithm proposed in section 5.3.2 is used for encrypting the images. The final encrypted images are shown in Figure 5.17(g,h,i) and their histograms are represented in Figure 5.17(j,k,l).

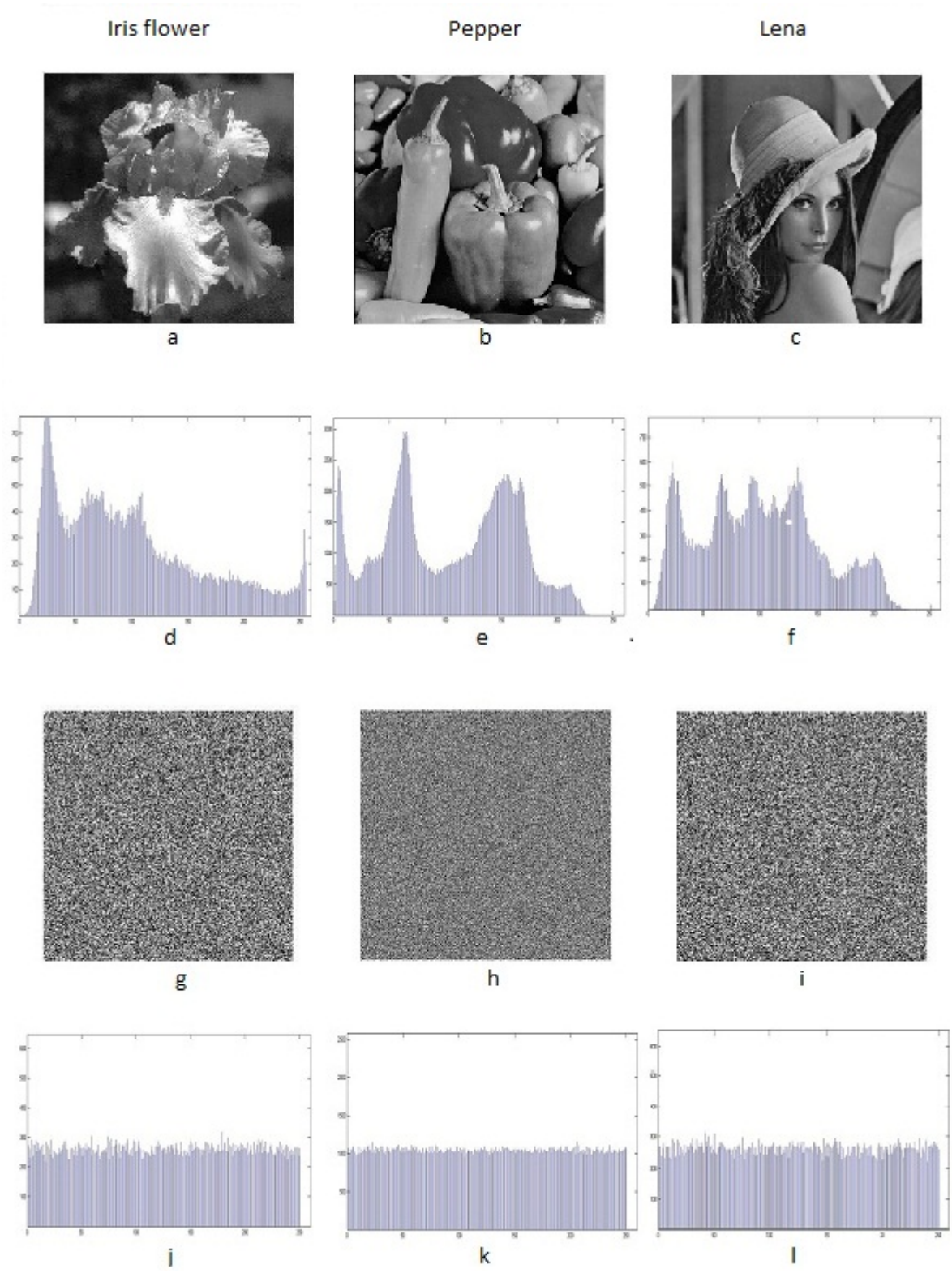

Figure 5.17: Additional simulation results

5.6 Summary

In this chapter we popose two stage encryption scheme of image. The first stage of encryption is done by modulo multiplication of pixel by integer chaotic sequence. The integer chaotic sequence is in the range of 1 to 256. The second stage of encryption is done by using a public key. The proposed scheme is proved to be quite efficient through cryptanalysis.

Chapter 6

Conclusion and Future Work

Chapter 6

Conclusion and Future Work

In this chapter, the overall contributions of the thesis are reported and future research problems are outlined for further investigation in the same or related topics. Different methods of generating self-invertible matrix are proposed. In order to make encryption robust, a sparse matrix is defined whose inverse can be obtained easily. The self-invertible matrix and sparse matrix are used as private keys in the asymmetric key cryptosystem. Several methods of generating orthonormal matrices are proposed and used in combination with the sparse matrix for encryption of images. Exponentiation operation on matrices is introduced and used for image encryption. Finally, a two-stage encryption technique based on chaotic sequence followed by any one of the cipher matrix techniques proposed in previous chapters is used to make encryption more robust. All the methods of encryption are validated using simulation technique.

The methods highlighted in this thesis are samples of generalised Hill cipher cryptosystem. In these methods, encryption is simpler in comparison to the decryption process because in decryption, one has to obtain the inverse of a large matrix in GF(p) and GF(p^n). Although the exponentiation operation makes the cryptosystem more secured, the computational complexity increases. Moreover,

the limitations of Hill cipher method also exist in these methods barring the two stage encryption technique discussed in chapter 5.

The future research areas are outlined here for further investigation -

- Generating self-invertible and othonormal matrices over Z_n and using them for encryption.
- Application of chinese remainder theorem in GF(p^n) for encryption.
- Encryption by representing data by variable radix number system and quantum number system.
- Introduction of error correcting codes for making cryptography more secured and immune to channel noise.

Bibliography

Bibliography

[1] W. Stallings, *Cryptography and Network Security*, 4th edition, Prentice Hall, 2005.

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

[3] Behrouz A.Forouzan, *Cryptography and Network Security*, Special Indian edition, Tata McGraw-Hill, 2006.

[4] Charles P. Pfleeger, Shari Lawrence Pfleeger, *Security in Computing*, 3rd edition, Prentice Hall Professional Technical Reference, 2003.

[5] R.E. Klima, N. Sigmon, E. Stitzinger, *Applications of Abstract Algebra with Maple*, CRC Press LLC, 1999.

[6] Damu Radhakrishnan, Yong Yuan, "A Fast RNS Galois Field multiplier", *IEEE International Symposium on Circuits and Systems*, New Orleans, LA, Vol. 4, pp. 2909 - 2912, 1-3 May 1990.

[7] F. Arguello, *Lehmer-based algorithm for computing inverses in Galois fields GF(*2^m*)*, ELECTRONICS LETTERS, Vol. 42, pp. 270 - 271, 2 March 2006.

[8] F. Ayoub, K. Singh, "Cryptographic techniques and network security", *IEE Proceedings Pt. F, No. 7*, Vol. 131, pp. 684 - 694, Dec 1984.

[9] Hui-Mei Chao, Chin-Ming Hsu, Shaou-Gang Miaou, "A Data-Hiding Technique with Authentication, Integration, and Confidentiality for Electronic Patient Records", *IEEE transactions on information technology in biomedicine*, Vol. 6, pp. 46 - 53, March 2002.

[10] Mohammad Ali Bani Younes , Aman Jantan, "Image Encryption Using Block-Based Transformation Algorithm", *IAENG International Journal of Computer Science*, 35:1, $IJCS\_35\_1\_03$, Advance online publication: 19 February 2008.

[11] Lala Krikor, Sami Baba, et al., "Image Encryption Using DCT and Stream Cipher", *European Journal of Scientific Research*, Vol. 32, No.1, pp. 48 - 58, 2009.

[12] Anastasios Tefas, Ioannis Pitas, "Image authentication using chaotic mixing systems", *IEEE International Symposium on Circuits and Systems*, Vol. 1, pp. 216 - 219, May 2000.

[13] Bibhudendra Acharya, Girija Sankar Rath, Sarat Kumar Patra, Saroj Kumar Panigrahy, "Novel methods of generating self-invertible matrix for hill cipher algorithm", *International Journal of Security*, Vol. 1 , pp. 14 - 21, 2007.

[14] Katsuki Kobayashi, Naofumi Takagi, Kazuyoshi Takagi, "An Algorithm for Inversion in GF(2^m) Suitable for Implementation Using a Polynomial Multiply Instruction on GF(2)", 18^{th} *IEEE Symposium on Computer Arithmetic(ARITH'07)*, Department of Information Engineering, Graduate School of Information Science, Nagoya University, pp. 105 - 112, 2007.

[15] Yi Shiung Yeh, Tzong-chenwu, Chin-ChenChang, "A new cryptosystem using Matrix transformation", 25^{th} *Annual 1991 IEEE International Carnahan*

Conference on Security Technology, Taipei, Taiwan, pp. 131 - 138, October 1991.

[16] Bao Ngoc Tran, Thuc Dinh Nguyen, "Modular Matrix Cipher and Its Application in Authentication Protocol", *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Phuket, pp. 318 - 323, August 2008.

[17] Yuen C., " Remarks on the Ordering of Walsh Functions", *IEEE Transactions on Computers*, C-21(1452), 1972.

[18] Uspensky J. V., Heaslet M. A., *"Elementary Number Theory"*, New York, McGraw-Hill, pp. 189 - 191, 1939.

[19] Apostol Tom M., *Introduction to Analytic Number Theory, Undergraduate Texts in Mathematics*, New York, Springer-Verlag, 1976.

[20] Arfken G. ,*Mathematical Methods for Physicists*, 2^{nd} ed. Academimic Press, Orlando, pp. 229 - 237, 1985.

[21] Hoffman K., Kunze R., *Linear Algebra*, Second edition, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[22] Nash J. C., *Compact Numerical Methods for Computers- Linear Algebra and Function Minimisation*, 2^{nd} edition, IOP Publishing Limited, Bristol, England, pp. 102 - 118, 1990.

[23] T. Habutsu, Y. Nishio, I. Sasase, et al., "A secret cryptosystem by iterating a chaotic map[A]", *EuroCrypt'91, Lecture Notes in Computer Science*, Vol. 547, Springer-Verlag, Berlin, pp. 127-140, 1991.

[24] Daan van den Berg, *"Discrete dynamical systems and the logistic map"*, Hogeschool van Amsterdam Riken brain science institute.

[25] Haojiang Gao, Yisheng Zhang, Shuyun Liang, Dequn Li, "A new chaotic algorithm for image encryption", *Chaos, Solitons and Fractals*, Vol. 29, pp. 393 - 399, July 2006.

[26] Ismail Amr Ismail, Mohammed Amin, Hossam Diab, "A Digital Image Encryption Algorithm Based on A Composition of Two Chaotic Logistic Maps", *International Journal of Network Security*, Vol.11, pp. 1 - 10, July 2010.

[27] F. Sun, S. Liu, Z. Li, Z. Lu, "A novel image encryption scheme based on spatial chaos map", *Chaos, Solitons and Fractals*, Vol. 38, pp. 631 - 640, 2008.

[28] Williams K. Pratt, Julius Kane, Harry C. Andrews, "Hadamard transform Image Coding", *Proceedings of the IEEE*, Vol. 57, pp. 58 - 68, 1969.

[29] Komo J.J., Yuan Chengchi, "Four level Hadamard transform", *Proceedings of the Twentieth Southeastern Symposium on System Theory*, pp. 188 - 191, 20-22 March 1988.

[30] Karagodin M.A., Osokin A.N., "Image compression by means of walsh transforms", *IEEE Trans. on Modern Techniques and Technologies(MTT)*, pp. 173 - 175, 2002.

[31] K.R. Rao, N. Ahmed, "Orthogonal transforms for digital signal processing", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, USA, Vol. 1, pp. 136 - 140, 12-14 April 1976.

[32] Acharya B., Panigrahy S. K., Patra S.K., Panda G., "Image Encryption Using Advanced Hill Cipher Algorithm", *International Journal of Recent Trends in Engineering*, Vol. 1, No. 1, May 2009.

[33] Jennifer Seberry, Beata J Wysocki, Tadeusz A Wysocki, "On some application of Hadamard matrices", *Metrika*, Vol.62, pp. 221 - 239, 2005.

[34] Debdeep Mukhopadhyay, Gaurav Sengar, Dipanwita Roy Chowdhury, "Hierarchical Verification of Galois Field Circuits", *IEEE transactions on computer-aided design of integrated circuits and systems*, Vol. 26, pp. 1893 - 1898, October 2007.

[35] B. Schneir, *"Applied Cryptography Protocols Algorithm and Source Code in C"* ,Wiley, NewYork, 1996.

[36] Xiang Zhou, Xiaohui Duan, Daoxian Wang, "A Semi-Fragile Watermark Scheme For Image Authentication", 10^{th} *International Multimedia Modelling Conference*, Brisbane, Australia, pp. 374 - 377, 5-7 January 2004.

[37] Anna Usakova, "DCT versus WHT for fragile image watermarking", *Journal of Electrical Engineering* , Vol. 55, No. 5-6, pp. 165 - 168, 2004.

[38] Philip Koopman, Tridib Chakravarty, "Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks", *International Conference on Dependable Systems and Networks*, pp. 145 - 154, 28 June-1 July 2004.

[39] Poulami Das, D.Bhattacharya, et al., "Discrete Fourier Transformation based Image Authentication technique", 8^{th} *IEEE International Conference on Cognitive Informatics*, Kowloon, Hong Kong, pp. 196 - 200, June 2009.

[40] Ye S., Zhicheng Zhou, Sun Q., Chang E.-C., "A Quantization based image authentication system", *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing*, Vol.2, pp. 955 - 959, 15-18 Dec. 2003.

[41] Marc Schneider, Shih-Fu Chang, "A Robust Content Based Digital Signature For Image Authentication", *IEEE Proceedings of the International Conference on Image Processing*, Lausanne, Vol. 3, pp. 227 - 230, 16-19 Sep. 1996.

[42] Wang Xiang, Lu Linzhang, "A Fast Sine Transform Algorithm For Toeplitz Matrices and Its Applications", *A Journal Of Chinese Universities*, Vol. 14, No.2, May 2005.

[43] Ali N Akansu, Richarda A Haddad, "On Asymmetrical Performance of Discrete Cosine Transform", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 38, Issue.1, pp. 154 - 156, January 1990.

[44] S. Morioka, Y. Katayama, T. Yamane, "Towards efficient verification of arithmetic algorithms over Galois fields GF(2^m)", *Proceedings of 13th International Conference on Computer Aided Verification*, Paris, France, pp. 465 477, 1822 July 2001.

[45] Huiyong Liao and Xiang-Gen Xia, "A Sharpened Dynamic Range of a Generalized Chinese Remainder Theorem for Multiple Integers", *IEEE Transactions On Information Theory*, Vol. 53, pp. 428 - 433, January 2007.

[46] Wen Tao Zhu, "Analyzing Euler-Fermat Theorem Based Multicast Key Distribution Schemes with Chinese Remainder Theorem", *IFIP International Conference on Network and Parallel Computing*, Shanghai, pp. 11 - 17, 1821 October 2008.

[47] Xiang-Gen Xia, Kejing Liu, "A Generalized CRT for residue sets with errors and its application in frequency determination from multiple sensors with low sampling rates", *IEEE Signal Processing Letters*, Vol. 12, No.11, pp. 768 - 771, November 2005.

[48] Yi-Shiung Yeh, Tzong-Chen Wu, Chin-Chen Chang, Yang W C, "A new cryptosystem using matrix transformation", *Proceedings of* 25^{th} *Annual 1991 IEEE International Carnahan Conference on Security Technology*, Taipei, pp. 131 - 138, 1-3 October 1991.

[49] Wenbo Mao, *Modern Cryptography-Theory and Practice*, Hewlett-Packard Company, Prentice Hall PTR, 2003.

[50] T. R. N. Rao, Kil-Hy Un Nam, "Private-key algebraic-code encryptions", *IEEE Transactions on Information Theory*, Vol. 35, Issue. 4, pp. 829 - 833, July 1989.

[51] W. F. Denny, "Encryptions using linear and nonlinear codes: implementation and security considerations", Ph.D. dissertation, The Center for Advanced Computer Studies, Univ. of South Western Louisiana, Lafayette, 1988.

[52] T. Hwang, T. R. N. Rao, "On the generation of large (s, s^{-1})pairs and permutation matrices over the binary field", *Tech. Rep. Centre for Advanced Computer Studies*, Univ. of Southwestern Louisiana, Lafayette, 1986.

[53] T. R. N. Rao, "Cryptosystems using algebraic codes", *Proc. of Int. Conf. on Computer Systems and Signal Processing*, Bangalore, India, Dec. 1984.

[54] R.C. Gonzalez, P. Wintz, *Digital Image processing*, 2^{nd} Edition, Addison-Wesley Publishing Co, Reading, MA, 1987.

[55] N. Ahmed, T. Natarajan, K. R. Rao, "Discrete Cosine Transform", *IEEE Transactions on computers*, Vol. C-23, No. 1, pp. 90 - 93, Dec. 1984.

[56] Kunitoshi Komatsu, Kaoru Sezaki, "Lossless 2D Discrete Walsh-Hadamard Transform", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Salt lake City, UT, Vol. 3, pp. 1917 - 1920, 711 May 2001.

[57] Jia Hou, Moon Ho Lee, "General Euler Hadamard /DFT/DCT Polynomial Function For Complex Signal Processing", 11^{th} *IEEE International Conference on High Performance Computing and Communications*, Seoul, pp. 533 - 538, 25-27 June 2009.

[58] Evaldo Gonsalves Pelaes, Yuzo lano, "Image coding using discrete sine transform with axis rotation", *IEEE Transactions on Consumer Electronics*, Vol. 44, Issue. 4, pp. 1284 - 1290, November 1998.

Dissemination of Work

Journals

[1] **S.K.Chhotaray**, J.Majhi, and G.S.Rath, Encryption by Hill cipher and by a novel method using Chinese Remainder Theorem in Galois field, *Int. J. Signal and Imaging Systems Engineering (IJSISE), Inderscience Publisher*, pp. 38-45, Vol. 6, no.1, 2013.

[2] **S.K.Chhotaray**, and G.S.Rath, Cryptography using Chaotic Neural Network, *Proceedings of Universal publishers*, ISBN-10 :1599428695, ISBN-13 :9781599428697, 2010.

Conferences

[1] **S.K.Chhotaray**, and G.S.Rath, Modified Hill cipher methods in Galois field, *International conference on Electronics Systems (ICES)*, NIT Rourkela, pp.419-423 , Jan. 9-11, 2011.

[2] **S.K.Chhotaray**, and G.S.Rath, Cryptography using Chaotic Neural Network, *International Conference on signals, Systems and Automation (ICSSA)*, GCET, V.V.Nagar, Gujarat, India, 28-29 December 2009.

[3] **S.K.Chhotaray**, and G.S.Rath, Encryption in Galois Field (GF), *National Conference on Emerging vistas of Technology in 21st Century*, PIET, Limbda, Gujarat, India, 11-12 September 11-12, 2009.

Sukant Kumar Chhotaray

Associate Professor

Department of Electronics & Communication Engineering

Sardar Vallabhbhai Patel Institute of Technology

Vasad 388 306

Gujarat, India.

Ph: +91-2692-274766 (O), +91-9426046558 (M)

e-mail: sukantchhotaray@gmail.com

Qualification

- Ph.D. (Continuing)

  NIT Rourkela

- M.Sc.Engg.(ESC)

  Sambalpur Universuty, Odisha

- B.Sc.Engg.(ETE)

  Sambalpur Universuty, Odisha, [First division]

- All India higher Secondary Examination (Science)

  Central Board of Secondary Education, New Delhi, India, [First division]

Publications

- 02 Journal Articles

- 03 Conference Papers

Date of Birth

February 25, 1959