CAPITAL UNIVERSITY OF SCIENCE AND
TECHNOLOGY, ISLAMABAD

Elliptic Curve Based Multi Secret Image Sharing

by

Mehwish Sehar

A thesis submitted in partial fulfillment for the
degree of Master of Philosophy

in the

Faculty of Computing
Department of Mathematics

2019

Dedicated to

My Parents

without their effort and prayers, I would never have reached so far

CERTIFICATE OF APPROVAL

Elliptic Curve Based Multi Secret Image Sharing

by

Mehwish Sehar

(MMT-171002)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Tayyab Kamran	QAU, Islamabad
(b)	Internal Examiner	Dr. Shafqat Hussain	CUST, Islamabad
(c)	Supervisor	Dr. Rashid Ali	CUST, Islamabad

Dr. Rashid Ali
Thesis Supervisor
May, 2019

Dr. Muhammad Sagheer
Head
Dept. of Mathematics
May, 2019

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
May, 2019

Author's Declaration

I, **Mehwish Sehar** hereby state that my M.Phil thesis titled "**Elliptic Curve Based Multi Secret Image Sharing**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my M.Phil Degree.

(Mehwish Sehar)
MMT-171002

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "*Elliptic Curve Based Multi Secret Image Sharing*" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of M.phil Degree, the University reserves the right to withdraw/revoke my M.Phil degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Mehwish Sehar)

MMT-171002

Acknowledgements

All praise be to **Almighty ALLAH** who has been bestowing me with his great bounties and enabled me to complete my dissertation.

I would like to thank my affectionate teachers, **Dr. Muhammad Sagheer, Dr. Abdul Rehman Kashif, Dr. Shafqat Hussain, Dr. M.Afzal and Dr. Rashid Ali** for their excellent teaching and support during these years.

I would like to express my special gratitude to my supervisor **Dr. Rashid Ali** for his patience with me and his guidance. He was a big support and motivation in the difficult times as he encouraged and helped a lot during research and writing of thesis. I feel really blessed and proud to be his student.

I am grateful to the management staff of Capital University of Science and Technology, Islamabad for providing a friendly environment for studies.

I am grateful to my parents and all the family members for their love and continuous support in achieving this target. I truly appreciate my father for all the prayers, without them, I would not have finished my degree.

I would also like to show my gratitude to my seniors specially, Sir Tahir for guidance and fellow students Saadia Noor, Saba, Sania Shah, and Sundas Iqbal for their encouragement along my studies. Their friendships are what I will miss the most and hope to keep forever. Especially, I would like to acknowledge Ms.Saadia Noor and appreciate her friendship and contribution. I remember the countless hours we were trying to learn and compile our LATEX files to create the presentable output. She always drove me towards my goal.

Finally, I am obliged to all people who pray for me, share their knowledge during my degree program and support me.

(Mehwish Sehar)

MMT-171002

Abstract

The significance of images and their sharing is increasing day by day. Their security is becoming an important issue while transferring over a public network. To protect images from hackers secret sharing is one of the best technique. The secret sharing is a way to share a secret with n participants and then setup is made for t or more number of participants who must contribute to revealing the secret. Here $t \leq n$ is known as a threshold which must be achieved for secret reconstruction.

In this thesis, we have proposed a scheme consisting of two phases for sharing multiple images secretly. Firstly $(2n, 2n)$ Secret Image Sharing phase generates $2n$ secret shares using boolean operation Exclusive OR (XOR) on $2n$ images. The secret cannot be releaved if $(2n - 1)$ shares are combined. Secondly, to make our scheme more secure we encrypt our shares by S-Box generated by the Elliptic Curve (EC) points.

Contents

List of Tables

Abbreviations

DES	Data Encryption Standard
AES	Advanced Encryption Standard
DLP	Discrete Log Problem
PKC	Public Key Cryptography
RSA	Rivest Shamir Adleman
ECC	Elliptic Curve Cryptography
SSS	Secret Sharing Scheme
PRN	Pseudo Random Numbers
ECDH	Elliptic Curve Diffie-Hellman
MSIS	Multi Secret Image Sharing
VSS	Visual Secret Sharing
EC	Elliptic Curve
SPN	Substitution Permutation Networks
SAC	Strict Avalanche Criteria
RGB	Read Green Blue

Symbols

$\mathcal{M}$	Plain text space
$\mathcal{C}$	Cipher text space
$\mathcal{K}$	Secret Key space
G	Group
$\mathbb{Z}$	Set of integers
$\mathbb{R}$	Set of real numbers
$\mathbb{Q}$	Rational numbers
$\mathbb{C}$	Complex numbers
$\mathcal{O}$	Point at infinity
$\mathbb{F}_p$	Galois Field
α	Combiner's private Key
β	Dealer's private Key

Chapter 1

Introduction

The secure transfer of private data over the public network is a big issue. In this context, there are many contributions to cryptography (a branch of cryptology). **Cryptography** is a science of secret communication which is used to alter the original message into unreadable form in the presence of a third-party over an insecure channel. We require a system for conversion of the original message into coded form. Such systems are recognized as cryptography. There is another branch of cryptology known as cryptanalysis. **Cryptanalysis** is an art of breaking a cryptosystem. Code breaking is usually considered to be a crime and there is a consideration that it should not be added as the main class of scientific discipline. Many researchers put their own contribution to the field of cryptanalysis. We may not judge the security of any cryptosystem without any attempt to break the cryptosystem.

The cryptographic schemes are classified into two classes based on key usage.

- Symmetric (Private) Key Cryptography
- Asymmetric (Public) Key Cryptography

The same key is used for encoding and decoding in the symmetric key cryptography by the sender and receiver. A major drawback of this system is that the

sender must transmit the key after encryption to the receiver for the decryption process. Some well known examples of symmetric key cryptography include Advanced Encryption Standard (AES) [1] and Data Encryption Standard (DES) [2]. This transfer of key is taken through a public channel. So, there may be a chance for compromising security. In 1976, Whitfield Diffie and Martin Hellman [3] introduced a new scheme known as asymmetric key cryptography to tackle this issue. Two different keys (encryption key and decryption key) are used in the asymmetric key cryptosystem. One key is used for the encoding process and other is used for the decoding process. Anyone can encrypt the data since the encryption key is public but only the person having the decryption key can decrypt the data because decryption key is private. Rivest Shamir Adleman (RSA) [4], ELGamal [5], and Elliptic Curve Cryptography (ECC) [6] are some examples of asymmetric cryptosystem. There may arise one question that how the asymmetric cryptosystems are formed? Mostly an answer to this question is given as this system is built by a common function known as a one-way function. Functions which can be evaluated in just one way but cannot be evaluated in the other direction without the special information known as Trapdoor function. Integer factorization [7] and Discrete Log Problem (DLP) [8] are the examples of trapdoor which are most commonly used in Public Key Cryptography (PKC). We will explain (DLP) in Chapter 2.

Some cryptographic techniques depending upon the encryption and decryption keys are used to protect the data for secure transmission. A cryptosystem is secure in a case if encryption and decryption keys are secure. Shamir [9] and Blackley [10] individually initiated the secret sharing scheme in 1979, to protect keys.

1.1 Why we Use Secret Sharing?

The significance of images and their sharing is increasing day by day. Their security is becoming an important issue while transferring images over a public

network. To protect images from hackers secret sharing is one of the best techniques. It was suggested with the inspiration of screening and securing private key in a cryptosystem. In 1979 A. Shamir [9] and G. Blakely [10] established the secret sharing and since then many secret sharing schemes were improved. Secret sharing can be organized into several kinds according to distinct requirements. It is classified into two classes based on a number of secrets to be shared, *i.e.* single secrets and multiple secrets. Moreover, same-weighted and multi-weighted shares are introduced by their share's capacity.

1.2 Literature Survey

Chen and Wu [11] suggested an efficient $(n, n+1)$ Multi Secret Image Sharing (MSIS) strategy based on XOR operations. In this strategy, n secret images are used to generate the $n+1$ shares. For decoding, $n+1$ share images are necessary to bring out the n secret images. In this technique, the capability to share out multiple secret images is gained. It uses only boolean computation for two meaningful images so, it cannot generate an arbitrary shared image.

Chen and Wu [12] proposed a secure boolean based secret image sharing scheme that generates random images from secret images by using random image generating function. To meet the random requirements a bit shift subfunction is used in the random image. From experimental results, it is analyzed that the CPU computational time is reduced to recover secret images. These images are shared and recovered in the same interval of time.

Shankar and Eswaran [13] have presented new Visual Secret Sharing (VSS) method for protecting images from attackers using Elliptic Curve Cryptography (ECC) with an optimization approach. In this manner, shares are produced from the secret image and every share is used for encoding and decoding process by means of ECC algorithm. In encoding step public key is generated arbitrarily and in decoding step secret key is optimally generated using optimization approach. Furthermore, PSNR values are acknowledged as a fitness value for the image.

Shankar et al. [14] presented an idea of (n, n) MSIS. According to this idea, n unrevealed images are encrypted into n shares. Elliptic curve cryptography is used to encrypt the multiple shares. High security and efficiency are achieved by using this idea for multiple hidden images and their shares.

Hayat and Azam [15] initiated two approaches for the construction of S-Box. At first, a total order on an Elliptic Curve (EC) is defined to obtain EC points. S-Box is generated by using x-coordinates of these points. Secondly, Pseudo Random Numbers (PRN) are gained to create an S-Box. The discussed approaches provide strong S-Box and PRN, optimal resistance against modern cryptosystems.

In this thesis, we have proposed a scheme consisting of two phases for sharing multiple images secretly. Firstly $(2n, 2n)$ Multi Secret Image Sharing (MSIS) phase generates $2n$ secret shares using XOR operation on $2n$ images. The secret cannot be releaved if $(2n - 1)$ shares are combined. Secondly, to make our scheme more secure we encrypt our shares by S-Box generated by the Elliptic Curve (EC) points [15].

1.3 Thesis Outline

This thesis consists of 4 chapters. A brief look towards these chapters is given below:

- **Chapter 1**

  In this Chapter, we have been discussed cryptology [16], it's two major sections cryptography and cryptanalysis. These two sections cover the cryptosystem properties, different types of keys and various cryptanalysis attacks. Furthermore, some basic definitions for image processing (encryption and decryption) and Group theory have been discussed. Moreover, secret sharing schemes, elliptic curve cryptography, and S-Box are explained.

- **Chapter 2**

  This Chapter will provide a detailed description of some basic concepts to understand the proposed scheme, as explained in Chapter 3. We shall describe cryptology, some basic definitions related to the Image processing, algebra and how secret sharing is important while transferring confidential images and it's various types. At the end of this chapter, a comprehensive explanation of elliptic curve cryptography and S-Box is given.

- **Chapter 3**

  We shall propose a scheme for encryption and decryption of $2n$ images. For this purpose, we shall explain our scheme by algorithms of encryption and decryption [17]. Additionally, we will use flow charts to clarify how our proposed scheme works. We illustrate our proposed scheme by taking a toy example of four multiple color images.

- **Chapter 4**

  Throughout this chapter, we shall analyze the performance of our scheme by applying some statistical tests including histogram, correlation and entropy test. We compare the results of original images and cipher images for analysis. Moreover, the security of the key for a secure cryptosystem is examined. For the implementation of the proposed scheme, "MATLAB" [18] is used.

Chapter 2

Preliminaries

2.1 Cryptology

The word **cryptology** [19] is originated from two Greek words **kryptos** (Hidden) and **logos** (words). Hence cryptology is a science for the safe and secure communication of data. It consists of two fields of study named:

1. **Cryptography**
2. **Cryptanalysis**

as shown in Figure 2.1.

2.1.1 Cryptography

Cryptography is the branch of cryptology that transforms the original message (audio, video or text) securely and it would be very difficult for an intruder to discover it's original meaning.

The sender transforms the original message or **Plaintext** ($\mathcal{M}$) into scrambled message or **Ciphertext** ($\mathcal{C}$). The process of transforming the '$\mathcal{M}$' into '$\mathcal{C}$' is known as encryption and process of transforming '$\mathcal{C}$' back into '$\mathcal{M}$' is known as

decryption. This process of encryption and decryption is done with the help of a **secret key** ($\mathcal{K}$) as shown in Figure 2.2.

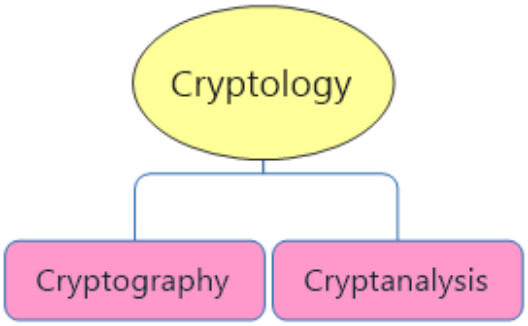

Figure 2.1: Types of Cryptology

In cryptography [20] usually the two characters, Alice and Bob are used. Alice (sender) wants to communicate with Bob (receiver) over the public network. The

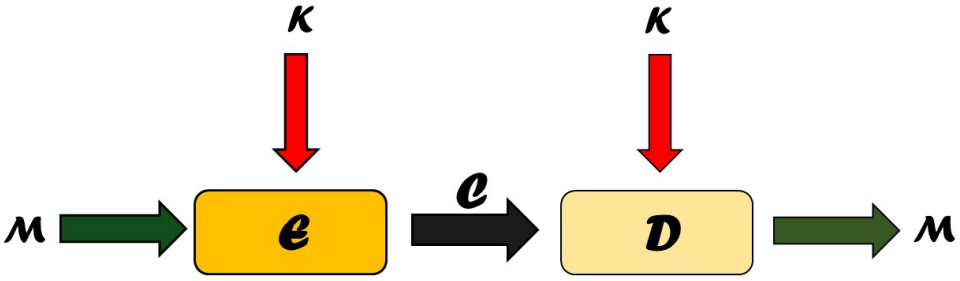

Figure 2.2: Cryptography

original message sent by Alice to Bob is known as plaintext. Plaintext is not sent to Bob in its original form but it is changed into a coded form called ciphertext. A ciphertext is a form of a message that is un-understandable for anyone, that's why it must be converted back into plaintext at the receiver's end. A process that converts plaintext into ciphertext is known as **encryption function** (encoding). The process of altering ciphertext into plaintext is named as **decryption function** (decoding). A key is the hypersensitive information used in encryption and decryption for the transformation of plaintext into ciphertext and vice versa. Authentication of a cryptosystem depends on key, therefore it must be kept secret. Some characteristics of cryptography [21] are described below:

Confidentiality

It ensures that only the sender and receiver have original information, and an unauthorized person cannot get the secret information. Suppose the two parties share secret information. The secret information is said to be confidential if the third party is unable to understand the original information even getting access to the secret information.

Data Integrity

It ensures that the transmitted information is not changed during the transmission over an unsecured channel, that is receiver can get the original information and nobody can change the encoded message except the sender and receiver.

Message Authentication

It confirms the identities of the senders and receivers. Let's consider that Alice and Bob want to communicate secretly. Message authentication ensures the identities of Alice and Bob. This property assures that an unauthorized person is not controlling their communication.

Certification

The certification can be defined as an information, that is transmitted by a trusted party or person.

Non-Repudiation

It states that the denial of the communicator at any stage is prevented that is the sender cannot deny at any stage about the delivery of any information. This property helps the receiver to trust the sender in any cryptosystem. The properties discussed above provide a highly trusted, secured and strong cryptosystem.

2.1.1.1 Types of Cryptography

A brief explanition for the types of cryptography as shown in the Figure 2.3 is given below.

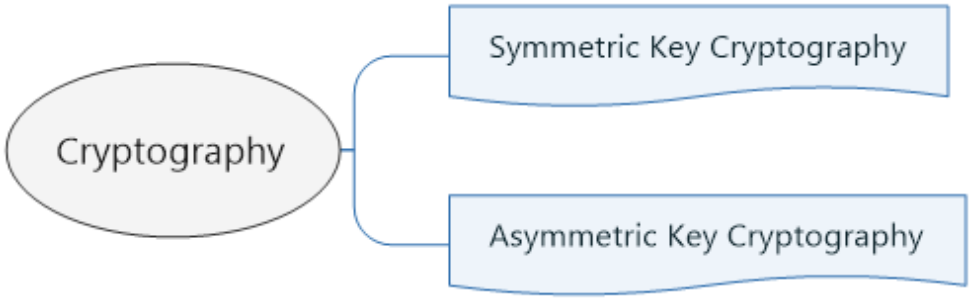

Figure 2.3: Types of Cryptography

2.1.1.2 Symmetric Key Cryptography

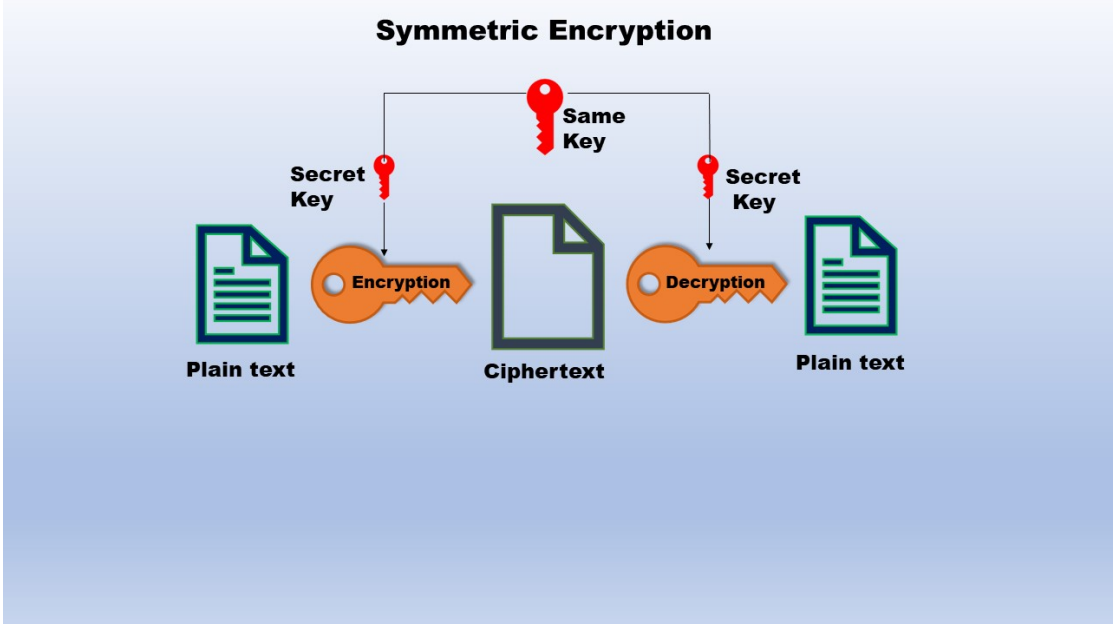

Figure 2.4: Symmetric Key

In symmetric key cryptography [22] a **single key** (private key) is used for both encryption and decryption. A typical symmetric key cryptographic model is shown in Figure 2.4. The private key $\mathcal{K}$ is unknown to adversary (attacker). Alternative name for symmetric key cryptography is the private or secret key cryptography.

2.1.1.3 Asymmetric Key Cryptography

A class of cryptography that depends upon the two keys (one is known to everybody called **public key** and the other which is kept secret known as **private key**) is specified as **asymmetric key cryptography** [23], descibed in Figure 2.5. One is used for encoding (encryption) and the other for decoding (decryption). Anyone can access public key easily but the private key can not be easily accessed. Since the public key is published publicly, therefore anyone with a copy of public key may encrypt the data. Secret key is used for decryption therefore, only the authorized person (who has the private key) can decode the data.

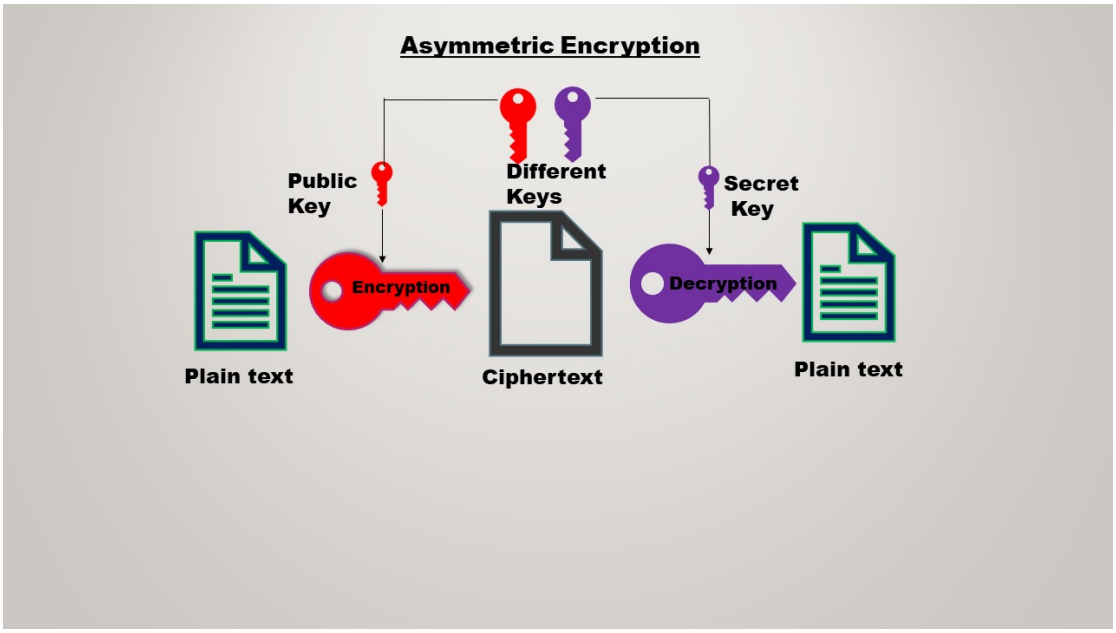

Figure 2.5: Asymmetric Key

The deduction of private key from public key is computationally infeasible, that is by knowing one key there are less chances to obtain the other key. In this scenario encoding is a public process on the other hand decoding is done by the users that only have the private key.

Examples of such a cryptosystem are RSA [4], ELGamal [24] and Diffie-Hellman key exchange [25].

2.1.2 Cryptanalysis

A process of acquiring plaintext from ciphertext without knowing the key is called cryptanalysis [26]. A person who takes the above process is called **cryptanalyst**. A cryptanalyst does this job if any of the four properties (confidentiality, data integrity, message authentication and non-repudiation) are found to be weak. If weakness is found then cryptosystem is said to be vulnerable to attack. Cryptanalysis is mainly used either for attacking a secret communication or to check the strength of cryptosystem.

Some of the attacks used for cryptanalysis are discussed below.

1. **Brute Force Attack**

   To reveal the plaintext from ciphertext, attacker randomly tries all the possible keys under this attack. The hardness of this attack is directly related with key size which is being used.

2. **Chosen Plaintext Attack**

   For attacking a cryptosystem there are many structures, chosen plaintext attack is one of them. Using this attack an attacker randomly picks some plaintext for encoding and get the corresponding ciphertext. An objective of this attack is to decrease the security of the encryption scheme for extracting more information about the ciphertext.

3. **Chosen Ciphertext Attack**

   It is the same scenario as the chosen plaintext, but it is applied on decryption function. An attacker arbitrary select some ciphertext and tries to get the corresponding plaintext. The purpose of using this attack is to obtain more information related to the plaintext.

4. **Ciphertext Attack**

   An attacker uses the ciphertext to attain the key or plaintext. For breaking the system letter's frequency can be used. Usually the attacker has no

information about the plaintext but he attacks the original message by using ciphertext attack.

5. **Known Plaintext Attack**

   In this attack a set of plaintext and its corresponding ciphertext is known to cryptanalyst. He uses previous information to decipher any further ciphertext or to figure out the key.

6. **Man-in-the Middle Attack**

   In this attack a hacker stays between the two parties who want to communicate secretly over the public network. Attacker completely controls both the sender and receiver's communication. In this attack, an attacker chooses two keys K_1 and K_2. The communication between the sender and receiver is fully controlled by the attacker in two phases. Initially, the sender encrypts his message with K_1 and send it to the receiver. Since there is an attacker between the sender and receiver therefore, an attacker gets the encrypted message. He/she can decrypt the encoded message. Secondly, an attacker encodes a message with the K_2 and sends this message to the receiver. He/she can also decode the reply obtained by the receiver. That's how an attacker holds the communication between the two parties without their knowledge.

2.2 Mathematical Background

Definition 2.2.1 (Pixel)

Pixel [27] is abbreviated from the word picture element. It is the smallest basic component of an image.

Definition 2.2.2 (Image Resolution)

Total number of pixels in an image describes the image resolution. It can be calculated by counting the total number of pixels. *i.e* an image having 2048 pixels in horizontal direction and 1536 pixels in the vertical direction (2048×1536)

contains 31457258 pixels or 3.1 megapixels. Hence the resolution of such image is 3.1 megapixels. Pixel count in a digital image is known as image resolution [28].

Definition 2.2.3 (Algorithm)

An algorithm is a step by step description for solving a problem. The problem is ultimately solved by following the steps defined in an algorithm.

Definition 2.2.4 (Image Encryption)

A method for converting the original image into the cipher image (by following the set of instructions explained in an algorithm) is called image encryption [29].

Definition 2.2.5 (Image Decryption)

The procedure of obtaining an original image from the cipher image is called decryption of image. Only an authorized user can decrypt [30] an image using a secret key.

Definition 2.2.6 (Group)

Let G be a non-empty set. G is said to be a group [31] under '$*$' where, '$*$' may be '$+$' or '$\cdot$' for any r, s and $t \in G$. If it satisfies the following properties:

a. Closure Property

G is said to be closed under '$*$' if $r * s \in G$

b. Associativity

G is said to be associative under '$*$' if for any r, s, $t \in G$ the following equality holds.

$(r * s) * t = r * (s * t)$

c. Identity Element

An element $e \in G$ is said to be an identity in G if

$e * r = r = r * e \ \forall \ r \in G$

d. Inverse

For any $r \in G$ if there exists $r^{'} \in G$ then $r^{'}$ is said to be inverse of r in G if

$r * r^{'} = i = r^{'} * r$

Definition 2.2.7 (Cyclic Group)

A Group G generated by a single element $h \in G$ is called cyclic group, where h is known as generator of G, denoted by $< h >$. Each element $i \in G$ is of the form h^s for some integer s. Furthermore, every cyclic group is an abelian group.

i.e. for all $i, j \in G$

$i * j = h^s h^t = h^{s+t} = h^{t+s} = h^t h^s = j * i \quad$ where, $\quad s, t \in \mathbb{Z}$

Definition 2.2.8 (Ring)

Let R be a non-empty set. R is said to be a ring [31] under '+' and '.' , if the following axioms are satisfied by R. for any r, s and $t \in R$

a. Closure Law

R is said to be closed under '+' if, $r + s \in R$

b. Associative Law

A ring R is said to be associative under '.' if for any r, s, $t \in R$ the following equality holds:

$r.(s.t) = (r.s).t$

c. Distributive Law

Left and right distributive law holds in R.

$r.(s + t) = r.s + r.t$

$(r + s).t = r.t + s.t$

Definition 2.2.9 (Commutative Ring)

If a ring R satisfies the commutative property under '.' then R is said to be commutative ring. *i.e* for any r, s, $t \in R$

$r \ . \ s = s \ . \ r$

$(\mathbb{Z}$, +, .), $(\mathbb{R}$, +, .) are some examples of commutative ring.

Definition 2.2.10 (Field)

A commutative ring R is called a field $\mathbb{F}$ [31] if an element $r \in R$ such that $r \neq 0$ forms a group under '.'.

$\mathbb{R}$, $\mathbb{C}$ and $\mathbb{Q}$ are examples of field.

Definition 2.2.11 (Finite Field)

A field containing a finite number of elements is called finite field. Galois field is an example of such field.

Definition 2.2.12 (Galois Field)

A field whose order is prime is known as Galois field [32]. A french mathematician named, Evaiste Galois proposed the Galois field in 1830. Finite field with p elements (where p is a prime number) is denoted by GF(p) or $\mathbb{F}_p$. It contains $\{0, 1, 2, .., p-1\}$ integers under modulo p.

2.3 Secret Sharing Scheme (SSS)

Secret sharing [33] is used to keep data secret. According to this scheme the data D is divided into different parts that is shared by a set U of trustees. The access structure [34] of SSS can be viewed by the following collection.

$\{ U' \subset U : \text{Data } D \text{ is stored by } U' \}$

An access structure of Shamir's threshold scheme is

$\{ U' \subset U : \mid U' \mid \geq k \}$ where, k is a positive integer.

There is a noticeable progress in data communication and computer networks. Networks read different forms of data. That's why it is becoming essential to save secret data from unauthorized persons. Data security is becoming an important problem. To secure confidential information secret sharing [35] is one of the best approach. In secret sharing scheme parts of a secret are allocated to the members of group. The reconstruction takes place only when a specific number of parts

bind with each other but the shares less than that specific number will not reveal any information or part of information.

In 1979 **Adi Shamir** the Israeli mathematician and **George Blackley** the American mathematician formulated the (t, n) threshold secret sharing scheme as shown in Figure 2.6. The whole process is controlled by a dealer and combiner. According to this scheme, there are n-number of participants. The secret is divided into n-shares by dealer. The secret can only rebuilt if t out of n shares are recombined by the combiner. The number t is specified by the dealer also note that $1 < t < n$.

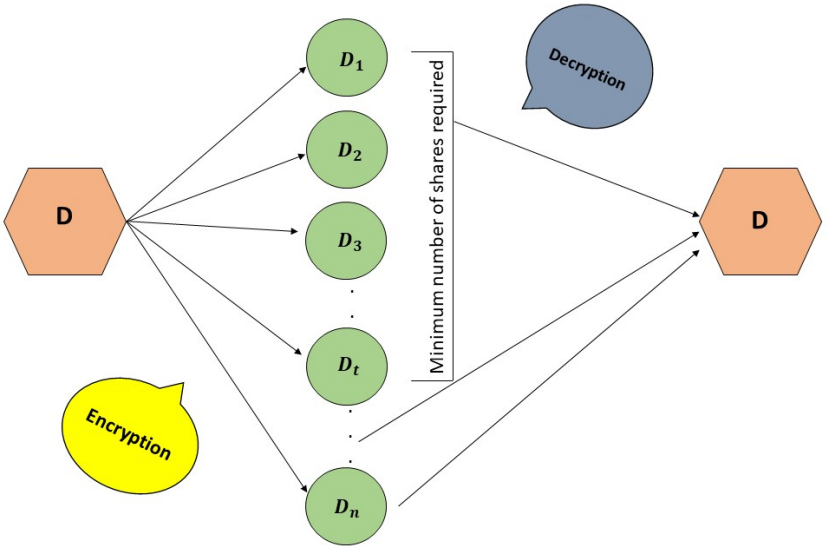

Figure 2.6: (t, n) threshold secret sharing scheme

Another kind of secret sharing scheme is (n, n) multi secret image sharing scheme introduced by **K. Shankar** and **G. Devika** in 2017. In this scheme a dealer takes n images from n users. After generating n shares of these images by a shared secret key. He/she distributes these shares to n participants. To reveal the secret these n participants submit their shares to combiner. Combiner can only transforms shares back into images only if he gets all the n shares from dealer by using the same key. Note that the images cannot be recovered even if $(n - 1)$ shares are combined by combiner.

Importance

For safe and flawless communication secret sharing is the best approach. It is extremely confidential and essential. Sensitive data including encryption keys, missiles launch codes and numbered bank accounts should be kept secret. It would be dangerous if it's security is compromised in any case. In threshold secret sharing approach a key is transmitted to several servers. It is necessary to recover the key. In security networks, secret sharing is also used to challenge the job of an eavesdropper. With the use of a various combination of share reconstruction, the security is enhanced.

2.4 Types of Secret Sharing

Secret sharing is categorized into diferent types, some of them is dicussed below.

1. **Ideal Secret Sharing**

   The distribution of a secret in the countable users in such a way that only the sanctioned users are combined to rebuild the secret from shares (these are same in size as of secret) is known as ideal secret sharing scheme [36].

2. **Non-perfect Secret Sharing**

   If some subset of participants cannot reconstruct the secret by having a little information about it then this type of secret sharing is said to be non-perfect [37].

3. **Protective Secret Sharing**

   If an attacker is succeeded to gain access towards the secret pieces of information and the participants accumulate them on unsafe computer servers. This process of reconstruction of uncompromised shares can be made practice. If the secret cannot be altered then this type of secret sharing is said to be protective secret sharing [38].

4. **Verifiable Secret Sharing**

   There may be a situation in which any participant tries to obtain the other shares by denying his own shares. A verifiable secret sharing scheme [39] ensures that the chance of denial for any participant about the shares is rare.

2.5 Elliptic Curve Cryptogrphy (ECC)

Elliptic curve [6] is defined by the equation,

$$y^2 = x^3 + sx + t \tag{2.1}$$

where s, t are constants, together with a point at infinity which is located at infinity and denoted by $\mathcal{O}$. And the non-singularity is ensured by

$$4s^3 + 27t^2 \neq 0. \tag{2.2}$$

In the above equations, both variables x, y are constants and the elements of some field $\mathbb{F}$. If we consider the field $\mathbb{F}$ over real numbers $\mathbb{R}$ then the variables and constants x, y, s, $t \in \mathbb{R}$ and the set of points that satisfies Equation (2.1), is denoted by $E_{\mathbb{R}}(s, t)$. Let's consider an elliptic curve,

$$y^2 = x^3 + x + 6. \tag{2.3}$$

In Equation (2.3), $s = 1$ and $t = 1$ and the variables $(x, y) \in \mathbb{R}$. The values of (x, y) satifying above equation are shown in Figure 2.7. Elliptic curve cryptography is based on elliptic curve theory [6]. It is an asymmetric key cryptography. It generates the keys that are efficient and smaller in size. The generation of key in an elliptic curve is based on EC properties instead of the product of very large prime numbers. Since it uses 160-bit Key size, therefore, a cryptosystem using ECC gains same security with smaller key size in comparison to other cryptosystem. That's how ECC makes the system secure with less computational cost and battery usage.

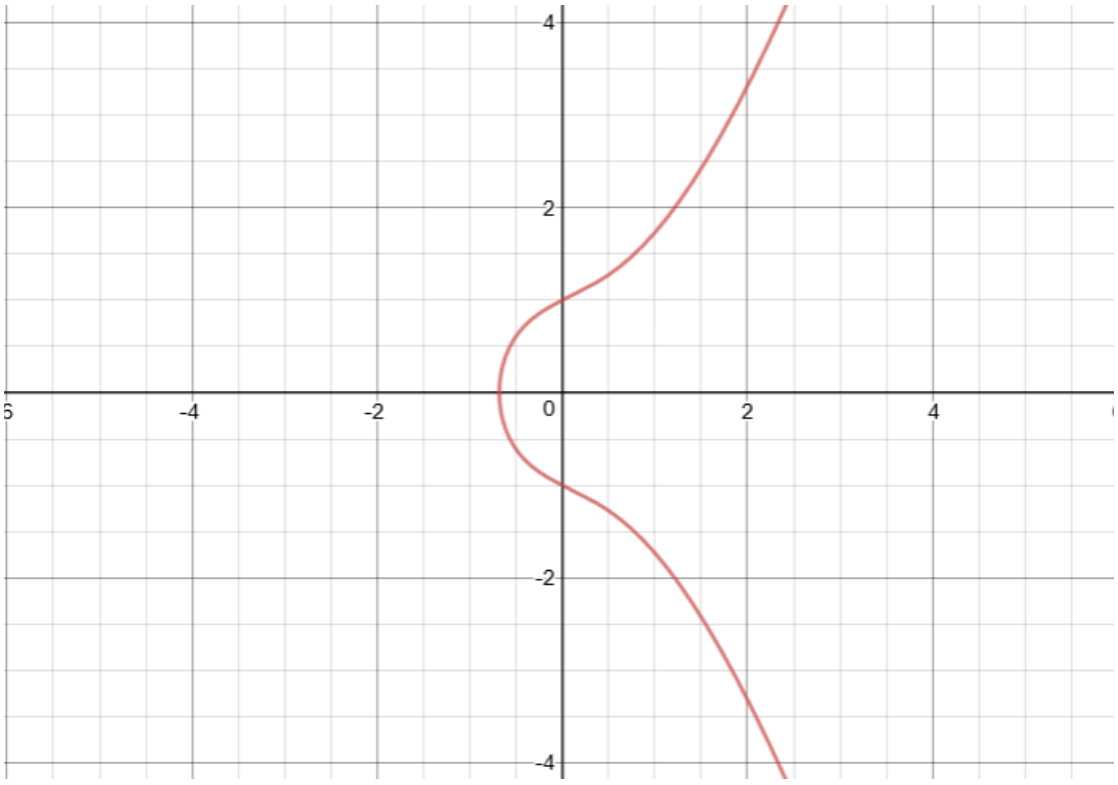

Figure 2.7: Graph of $E(1,1)$ over $\mathbb{R}$

It is observed from many researches on ECC that the challenge to solve an elliptic curve discrete logarithm problem is exponentially tough with respect to the key size being used. Due to this property, ECC is becoming a platform for encryption and decryption process in comparison with other cryptographic approaches which are linearly or sub exponentially hard. In the year 1985 **Neal Koblitz** and **Victor S. Miller** established ECC. It is widely trusted and used in most of encoding and decoding schemes.

The ECC is a practical and secure skill to be employed in controlled applications. For producing a continuous cryptographic curve, we use generating curves technique which goes through different algorithms and process. ECC is recommended over RSA since it provides the same security as that of RSA but with smaller key space.

Elliptic curve cryptographic schemes are reliable and reduce computational cost. Discrete logarithm can be attacked using several attacks. If at least 1024-bit field is

used, then this attack cannot be applied. Elliptic curve provides the same security by using a field of 160-bits that's why ECC saves time and space.

2.6 Group Operations on Elliptic Curve

In this section, we will explain some mathematical operations used in image encryption and decryption based on elliptic curve cryptography [40]. Geometry of elliptic curve point addition is shown in Figure 2.8. Let's take two points K and L having coordinates u, v.

Point Addition

$K(u_1, v_1) + L(u_2, v_2) = M(u_3, v_3)$

Let's consider some basic concepts before adding two different points using ECC.

1. **Identity Element**

   The point at infinity $\mathcal{O}$ is called an identity element, that is $K + \mathcal{O} = K$.

2. **Negative of a Point**

   Negative of y-coordinate of a point is known as it's negative.

   i.e. Negative of $K(u_1, v_1) = K(u_1, -v_1)$

Steps for Adding Two Different Points

Let K and L be two different points, we can add them by following procedure to obtain addition of distinct points.

- Draw a straight line through the points K and L.
- A point M is obtained by the intersection of straight line and elliptic curve.
- Take negative of M.
- This negative of M is our desired addition of two distinct points.

Note that an Identity $\mathcal{O}$ is obtained by $M + (-M)$.

Point Doubling

If a point is added to itself, this process is known as point doubling. The following steps are used for point doubling.

- Draw a tangent line through the point K.
- Tangent line intersects an elliptic curve at a point N.
- Find negative of N.
- The negative of N is our desired point.

Point Multiplication

Base point is repeatedly added in elliptic curve point multiplication,

that is $nK = \overbrace{K + K + \ldots + K}^{(n\ times)}$; where $n = 1, 2, 3, ...$

Geometrical Representation and Formulation for Group Operations on Elliptic Curve

Using the two distinct points $K(2, 4), L(-1, 2)$ of Equation (2.3), the point addition is graphically shown in Figure 2.8. The defining formulae for adding points on an elliptic curve is given as,

$$T : y = bx + c, \tag{2.4}$$

where, T is a line through the points K and L and b is the slope of line T. Slope 'b' of line in Equation (2.4) is defined as follows:

$$b = \begin{cases} \frac{v_2 - v_1}{u_2 - u_1}, & \text{if points are distinct} \\ \frac{3u_1^2 + s}{2v_1}, & \text{for point doubling.} \end{cases}$$

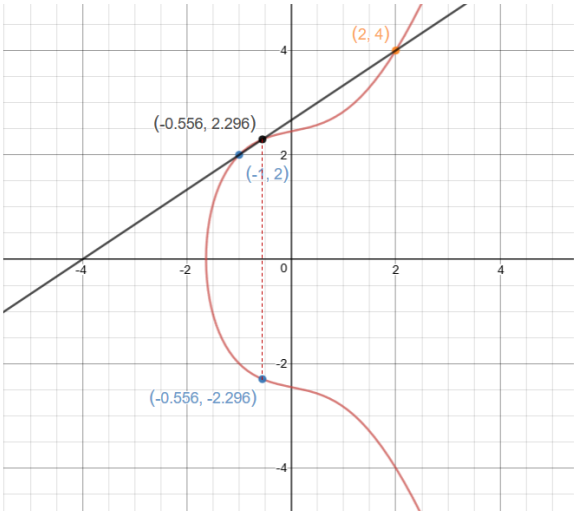

Figure 2.8: EC Point Addition

Mostly group is defined on elliptic curve over the finite field [41] $\mathbb{F}_p$ *i.e.* $E_{\mathbb{F}_p}(s,t)$. Elliptic curve was introduced by **Neal Koblitz and Victor Miller in 1985**. An equation of elliptic curve over $GF(p)$ is defined in Equation (2.5)

$$y^2 = (x^3 + sx + t) \mod p \tag{2.5}$$

with point at infinity $\mathcal{O}$. This is a non-singular curve and has distinct roots which satifies the equation $4s^3 + 27t^2 \neq 0$, where x, y, s, $t \in \mathbb{F}_p$. The operations for point addition discussed above are same in finite field $\mathbb{F}_p$. We just have to do all working in modulo p.

2.7 Elliptic Curve Discrete Log Problem

As elliptic curves over finite field $\mathbb{F}_p$ generates cyclic group. We can write,

$$\overbrace{K + K + \ldots + K}^{(n\ times)} = nK = L$$

for $\mathbb{F}$ and for points K and L in additive group. We can compute L easily by examining the above equation, if we know n and K. Yet it is tough to find n even we know about K and L, this is called elliptic curve discrete log problem [8],

that is $n = log_K L$.

2.8 Elliptic Curve Diffie-Hellman Key Exchange Protocol

Elliptic Curve Diffie-Hellman key exchange [3] allows the dealer and combiner to exchange the secret key over an unsecure network as in Figure 2.9. Once this key is being shifted it becomes symmetric key. This procedure is described into

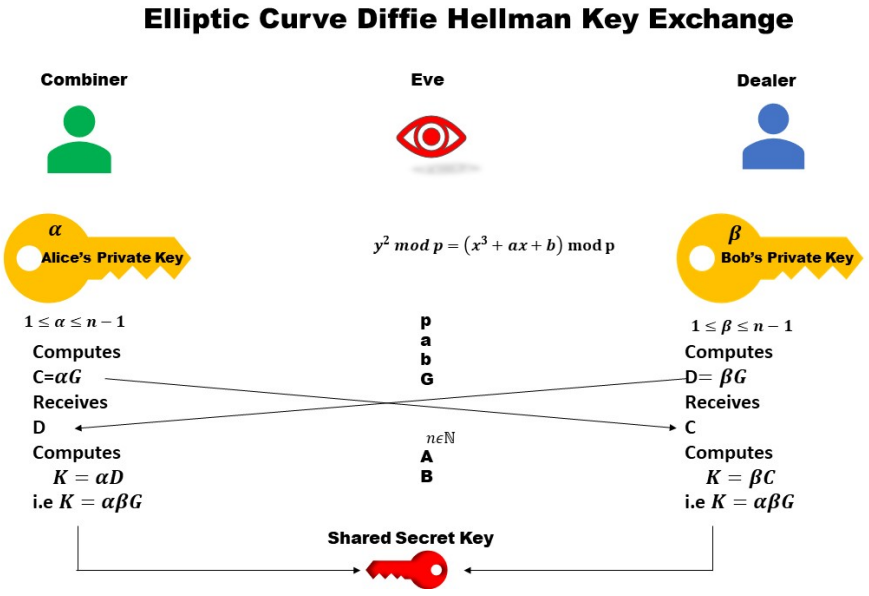

Figure 2.9: ECDH key exchange

following steps:

- Firstly, combiner and dealer decide a specific elliptic curve, prime number p and a base point G from the decided EC points. All the parameters discussed above should be kept public.

- Combiner picks a private key α such that $1 \leq \alpha \leq n-1$ and computes $C = \alpha G$ and transmits C to dealer.
- Dealer picks a private key β such that $1 \leq \beta \leq n-1$ and computes $D = \beta G$ and transmits D to combiner.
- Lastly, combiner and dealer both calculates the private shared key (symmetric key) $K(a_1, b_1)$ after computing $a_1 = \beta\ C$ and $b_1 = \alpha\ D$.

2.9 S-Box

A table containing $u \times v$ mapping of the form $\{0,1\}^u \longrightarrow \{0,1\}^v$ is called S-Box (Subsitution Box) [42], where u and v are non-negative integers.

- **Boolean Function**

  A function whose independent variables are deduced from $\{0,1\}^u$ to $\{0,1\}$ to set, is known as boolean function. There are two independent variables in a boolean function.

  or

  A mapping $f : \{0,1\}^u \longrightarrow \{0,1\}$ is said to be boolean function. where u is a non-negative integer.

- **Truth Table of a Boolean Function**

  A table expressing the boolean reperesentation of a logic gate (AND, OR, NOR, NAND, etc) is known as truth table of a boolean function.

A Brief Description of S-Box

Block cipher and Substitution Permutation Networks (SPN) [43] are basically the two extensions of Shannon's theory of confusion and diffusion. Usually, a block of plaintext and key is taken as input and several rounds of Substitution Box (S-Box) or Permutation Box (P-Box) is applied to input for acquiring the desired

ciphertext. Inverse S-Box or P-Box are implemented for the decryption process with the same key, Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are two examples of SPN.

Stream Cipher and Block Cipher are the two major classes of the symmetric key cryptosystem. The whole block of data is encrypted into coded form at simultaneously by using block cipher with a secret key. Two components of block cipher are block size and key size.

It has a basic role in the contemporary scheme of iterative block ciphers. Classicaly, it suppress a relationship between the key and ciphertext while dealing with iterative block ciphers.

Categories of S-Box

S-Box is categorized into three types, **(i)** Straight S-Box **(ii)** Expanded S-Box **(iii)** Compressed S-Box.

i. Straight S-Box

S-Box which uses the same data bits for both input and output is known as straight S-Box. It is the simplest and easiest type of S-Box. AES is an example of such S-Box.

ii. Expanded S-Box

It receives fewer bits as input and generates an output of more data bits. By duplicating some input or output bits such S-Box can be constructed.

iii. Compressed S-Box

A design of S-Box which takes in more bits and output fewer bits is called compressed S-Box. An excellent example of compressed S-Box is DES in which 6 input bits are taken as one input block and 4 bits in one block are returned as output block.

Standards for an Ideal S-Box

An ideal S-Box [44] should be very simple in its design, supports encryption and decryption efficiency secured against known plaintext attacks. Some of the desirable properties for an ideal S-Box approved by National Institute of Science and Technology (NIST) are,

- **Balanced S-Box**

  Equal number of zeros and ones in the truth table refers to a balanced S-Box.

- **Non-Linearity**

  The distance between the functions and set of all affine function is called Non-linearity of an S-Box. S-Box with high Non-linearity is a good resistor of linear attacks.

- **Hamming Weight**

  The number of ones in binary sequence specifies to hamming weight.

- **Strict Avalanche Criteria (SAC)**

  Alteration in one bit of S-Box results in a change of more than half of its output bits.

- **Higher Irder SAC**

  Change in more than one bits is defined as higher order strict avalanche criteria.

- **Propagation Criteria**

  Propagation criteria is obtained by a combination of SAC and higher order SAC.

Chapter 3

Secure and Efficient Multi Secret Image Sharing (MSIS) Scheme Based on Boolean Operation XOR and S-Box Encryption

In this chapter, we will analyze and improve a scheme based on boolean operation (XOR) and ECC encryption proposed by **Shankar *et al.*** [14] in 2017. In the proposed scheme basic idea of share generation for $2n$ images is taken from the paper discussed above. Secondly, to make our scheme more secure we encrypt our shares by using S-Box generated by elliptic curve points taken from EC point addition.

3.1 (MSIS) Scheme Based on Boolean Operation XOR and Elliptic Curve Encryption

This scheme is recognized as (n, n) Multi Secret Image Sharing (MSIS) because n multiple secret shares are generated from the n multiple secret images. The

original images can be revealed if n shares are recombined. This scheme is different from Shamir's (t, n) secret sharing scheme [45] because in reconstruction phase the secret can be revealed only, if all the n shares are combined instead of any t shares, and secret cannot be revealed if $(n-1)$ shares are combined. If an intruder collects all the shares then there is a possibility of retrieval of original shares. Dealer and combiner controls the process of division and reconstruction of secure images in this scheme. Figure 3.1 represents this scheme in which a dealer collects n different images and generate n shares of these images. Dealer uses the secret shared key generated by combiner and dealer using Diffie-Hellman key exchange for encryption of shares. Combiner at the reconstruction phase uses the same key for recovering the secret images.

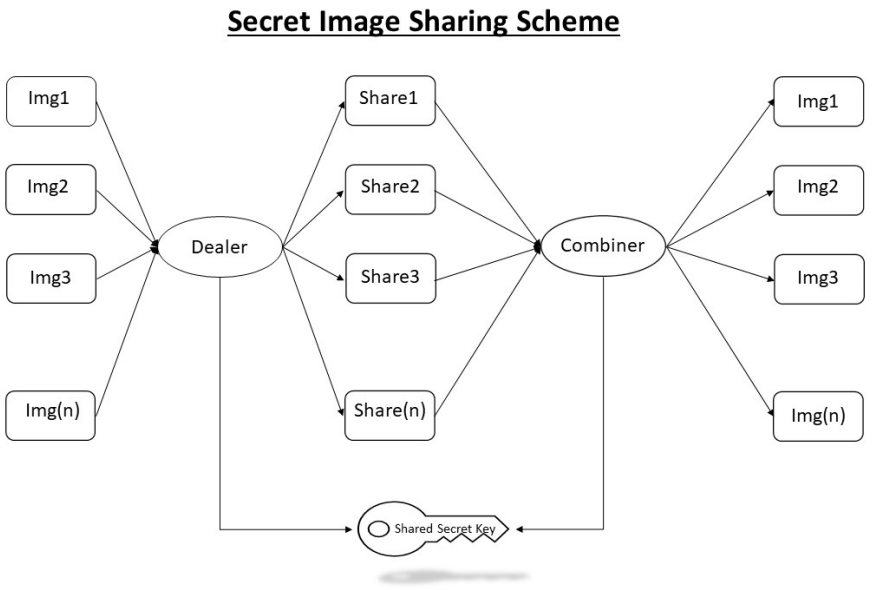

Figure 3.1: (n, n) Secret Image Sharing Scheme

A two phase encoding procedure is taken throughout this scheme. At first, images are encoded while share construction and then these shares are encoded by ECC encryption. Steps for the encoding process are given below.

- **Encryption Algorithm for (n,n) Secret Image Sharing Scheme**

Input: n secret RGB images $\{P_1, P_2, P_3, \ldots, P_n\}$.

Output: n cipher RGB images $\{C_1, C_2, C_3, \ldots, C_n\}$.

Step 1

Use "MATLAB" to read the n secret RGB images.

Step 2

Exract the colour components of each image as,

$P_1 = P_{R1}, P_{G1}, P_{B1}$

$P_2 = P_{R2}, P_{G2}, P_{B2}$

$P_3 = P_{R3}, P_{G3}, P_{B3}$

$\vdots$

$P_n = P_{Rn}, P_{Gn}, P_{Bn}$

Step 3

Combine the red, green and blue colour components of all the images as follows.

$P_{RN} = (P_{R1} \oplus P_{R2} \oplus P_{R3} \oplus \ldots \oplus P_{R(n-1)} \oplus P_{Rn})$

$P_{GN} = (P_{G1} \oplus P_{G2} \oplus P_{G3} \oplus \ldots \oplus P_{G(n-1)} \oplus P_{Gn})$

$P_{BN} = (P_{B1} \oplus P_{B2} \oplus P_{B3} \oplus \ldots \oplus P_{B(n-1)} \oplus P_{Bn})$

$M_1 = (P_{RN},\ P_{GN},\ P_{BN})$

Step 4

Use bit complement operation on M_1 for obtaining M_2 *i.e.*

$M_2 = \sim M_1$

Step 5

Apply bit XOR (boolean operation) on RGB components of n multiple images and M_2 for acquiring n multiple shares *i.e.*

$Share_1 = (P_{R1} \oplus M_2, P_{G1} \oplus M_2, P_{B1} \oplus M_2)$

$Share_2 = (P_{R2} \oplus M_2, P_{G2} \oplus M_2, P_{B2} \oplus M_2)$

$Share_3 = (P_{R3} \oplus M_2, P_{G3} \oplus M_2, P_{B3} \oplus M_2)$

$\vdots$

$Share_n = (P_{Rn} \oplus M_2, P_{Gn} \oplus M_2, P_{Bn} \oplus M_2)$

Step 6

For making this scheme more secure, an ECC [46] based encoding method is used. This coding procedure is based on the EC equation:

$$y^2 = x^3 + mx + n \mod p \tag{3.1}$$

where m and n are integers and p is prime.

Step 7

The points of an Elliptic Curve (EC) defined in Equation (3.1) are of the form $P = (c, d)$, and G is it's base point.

Step 8

Elliptic curve point doubling process is used to find the points of EC by using two public keys G and H where $H = \gamma G$ and γ is the private key.

Step 9

Convert each share into 8×8 block. The total number of blocks represents the total number of pixels (j, k). The rows and columns of each block of share are represented by j and k. The consecutive two pixels of the image are taken as input each time *i.e.* the pixels $D_x(j, k)$ and $D_y(j+1, k)$ are mapped to the ECC points:

$$G_1 = \gamma G$$

$$G_2 = (D_x, D_y) + G_1$$

Step 10

To encrypt multiple shares into multiple secret images repeat Step 6 to Step 9. In this way each share is converted into cipher image *i.e.* $\{C_1, C_2, C_3, \ldots, C_n\}$.

- **Decryption Algorithm for (n,n) Secret Image Sharing Scheme**

Input: n cipher RGB images $\{C_1, C_2, C_3, \ldots, C_n\}$.

Output: n secret RGB images $\{P_1, P_2, P_3, \ldots, P_n\}$.

Step 1

Read the cipher images $\{C_1, C_2, C_3, \ldots, C_n\}$ by using "MATLAB".

Step 2

To decode the multiple cipher images the point $G_2 = \gamma G_1$ is used. Shares can be recovered from cipher images by the following transformation.

$$Q = G_2 - G_1$$

$\Rightarrow Q$ represents the decrypted shares. Repeat this method for getting multiple shares.

Step 3

Combine all the shares of Step 2 by using XOR operation to obtain M_1 *i.e.*

$M_1 = Share_1 \oplus Share_2 \oplus Share_3 \oplus \ldots \oplus Share_n$

Step 4

Apply bit complement operation on M_1 to get M_2.

$M_2 = \sim M_1$

Step 5

The following method is taken to recover the secret multiple images.

$P_1 = Share_1 \oplus M_2$

$P_2 = Share_2 \oplus M_2$

$P_3 = Share_3 \oplus M_2$

$\vdots$

$P_n = Share_n \oplus M_2$

Note

The scheme discussed in Section 3.1 was claimed for (n, n) multi secret RGB images but after analyzing this scheme it is concluded that it works only for even number $(2n)$ of images. Moreover, the computational cost is increased while encrypting multiple shares by using ECC encryption technique. Therefore, we are improving this scheme by proposing a scheme that works with the even number $2n$ of images in Section 3.3. In the proposed scheme, we will use ECC to construct an S-Box for encryption. Using the S-Box encryption same security is achieved with higher efficiency due to S-Box encryption.

3.2 The Proposed Methodology

Secret sharing is a process for the allotment of secret shares [47] among the contributors. The secret image is encoded into multiple shares which are distributed in contributors. In the reconstruction phase single share is considered worthless. The secret image can be reconstructed only when specified shares are combined. In most of the secret sharing schemes, a single secret image is being shared. With the rapid growth on the internet, it is becoming essential to transfer multiple images at a time without compromising the security. Hence, we are proposing $(2n, 2n)$ (MSIS) scheme consisting of two phases. In our scheme initially, $2n$ images are encoded into $2n$ shares using boolean operation XOR. To enhance the security these shares are further encrypted using the S-Box generated by right side values of K elliptic curve points. In the suggested scheme as described in Figure 3.2,

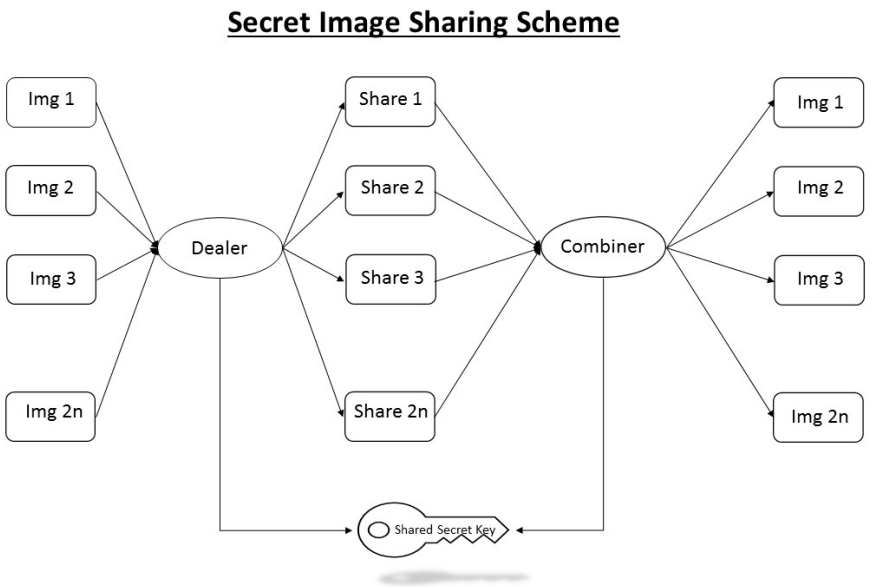

Figure 3.2: $(2n, 2n)$ Secret Image Sharing Scheme

dealer has $2n$ distinct images and encryption is performed by the dealer using secret shared key (shared by elliptic curve Diffie-Hellman key exchange protocol between the dealer and combiner). Combiner uses the same secret shared key for decryption (to get original images).

3.3 Algorithm for Proposed Scheme

- **The proposed $(2n, 2n)$ secret image sharing procedure for encryption**

  Before getting a look into the whole procedure let's understand with the help of flow chart displayed in Figure 3.3.

  Flow Chart for Encryption of $2n$ Images

  Multiple shares of multiple images are generated by taking the following steps.

Input: K secret colour images I_K ,

where, $K = 1, 2, 3, \ldots, 2n$

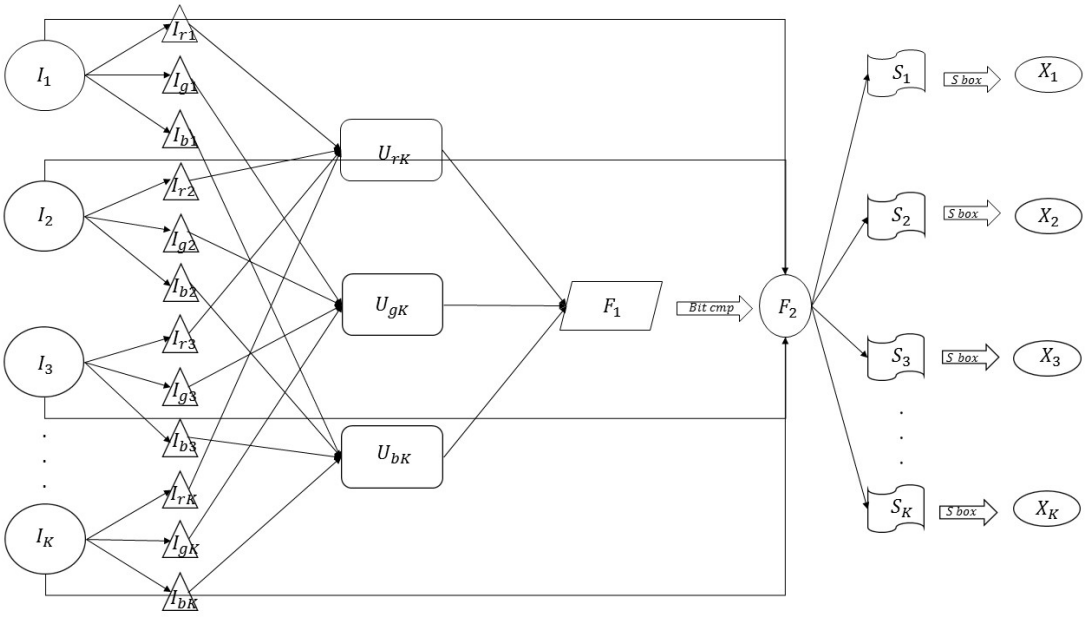

Figure 3.3: Encryption of $2n$ Images

Output: K cipher colour images T_K.

- **Generation of Shares**

Step 1

Read the K secret colour images by using "MATLAB" where, $K = 1, 2, 3, \ldots, 2n$.

Step 2

Choose the colour components from all the secret images.

Secret colour images $= \left\{ I_1, I_2, \ldots, I_K \right\}$. The RGB components of I_K images are given below.

$I_K = (I_{r_K}, I_{g_K}, I_{b_K})$

Step 3

Apply the boolean operation XOR on the seperate colour components to get F_1.

$F_1 = (U_{r_K},\ U_{g_K},\ U_{b_K})$

step 4

Apply bit complement operation on F_1 to get F_2 as follows.

$F_2 = \sim F_1$

Step 5

Produce the RGB shares of all the plain images by the following steps.

$Share_1 = (I_1 \oplus F_2)$

$Share_2 = (I_2 \oplus F_2)$

$\vdots$

$Share_K = (I_K \oplus F_2)$

- **S-Box Encryption of Shares**

Since an intruder cannot approach the secret with a single share but there are chances that a hacker can get the secret. If he succeed to obtain all the shares. We further encrypt the shares using the S-Box, for making our scheme more secure. S-Box generated by elliptic curve points is shown in Table 3.1.

Construction of S-Box:

S-Box can be generated by the following steps.

Step 1

Choose an elliptic curve as defined in Equation (2.4).

Step 2

Apply the ECDH key exchange protocol explained in Section 2.8 to obtain shared secret key $K(a_1, b_1)$.

Step 3

Use elliptic curve point doubling described in Section 2.6 by taking $K(a_1, b_1)$ to get $2K(a_2, b_2)$

where, $2K = K + K$.

Table 3.1: **S-Box**

83	27	146	179	173	18	50	85	139	71	47	121	205	67	137	142
195	96	123	181	113	12	84	109	193	213	102	135	167	2	172	95
53	30	152	29	224	41	206	178	253	44	68	151	97	242	91	217
106	93	7	82	166	145	78	153	5	251	13	215	150	231	34	25
127	117	77	116	161	57	21	255	216	37	16	65	129	110	107	103
158	39	125	191	247	200	169	52	42	51	243	118	69	249	240	63
144	235	228	75	164	62	55	239	159	136	192	236	3	31	70	126
120	59	199	208	218	114	250	128	94	61	198	73	38	49	171	112
119	175	248	10	246	201	148	54	220	187	9	130	226	180	185	188
124	45	98	56	4	14	225	168	223	122	24	170	87	76	0	46
6	86	92	115	165	189	232	60	207	101	140	238	154	32	214	141
66	197	19	99	194	58	43	28	177	26	183	254	176	22	233	36
211	35	212	89	202	244	160	227	15	210	40	20	149	33	17	143
237	88	80	245	229	186	72	8	11	138	132	111	1	219	108	133
163	230	241	64	157	155	74	204	23	156	105	162	90	221	182	209
190	131	81	234	174	104	100	222	196	79	252	147	184	137	48	203

Step 4

For generating multiple distinct elliptic curve points $nK(a_i, b_i)$, $i = 1, 2, ..., 256$ use elliptic curve point addition as shown in Figure 2.8.

Mathematically, $nK = K + (n-1)K$ for $n = 1, 2, \ldots, 256$.

Step 5

Select b_i such that $b_i \neq b_j$.

Step 6

Use the following transformation at "MATLAB" to convet b_i into 8-bit integers

$$J = 256 \times [(X - n) \div D],$$

where, $X = b_i \div 10,000, \quad D = m - n, \quad m = max(X), \quad n = min(X).$

Step 7

Store all the values of J row-wise to get S-Box as shown in Table 3.1.

Step 8

Use this S-Box as a look up table to encrypt each pixel of all secret shares by using "MATLAB".

The above encryption procedure is adopted for all shares.

- **The Proposed $(2n, 2n)$ Secret Image Sharing Procedure for Procedure for Decryption**

Decryption flow map as shown in the Figure 3.4, will assist in understanding the proposed recovery algorithm.

For recovering the original images from secret shares, steps of the algorithm are stated below.

Input: K cipher images T_K.

Output: K original images (secret colour images) I_K,

where $K = 1, 2, 3, \ldots, 2n$.

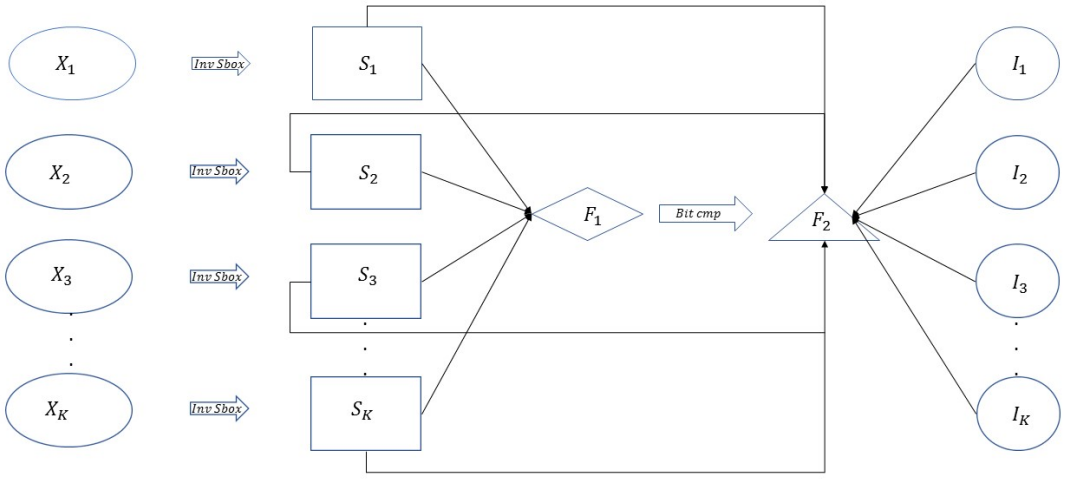

Figure 3.4: Decryption of Cipher Images

- **Recovery of Original Images**

Original images can be recovered by the following steps.

Step 1

Use "MATLAB" to read the multiple encoded images T_K .

Step 2

Apply inverse S-Box shown in Table 3.2 generated by inverse of S-Box on each encoded image T_K in order to obtain the multiple shares.

Step 3

Combine all the shares that are obtained in Step 2 by using XOR operation and compute F_1.

Step 4

Determine the function F_2 after taking bit complement of F_1

that is, $F_2 = \sim F_1$.

Step 5

Adopt the following process to recover multiple plain images.

$I_1 = Share_1 \oplus F_2$

$I_2 = Share_2 \oplus F_2$

$\vdots$

$I_K = Share_K \oplus F_2.$

Table 3.2: **Inverse S-Box**

158	220	29	108	148	56	160	50	215	138	131	216	21	58	149	200
74	206	5	178	203	70	189	232	154	63	185	1	183	35	33	109
173	205	62	193	191	73	124	81	202	37	88	182	41	145	159	10
254	125	6	89	87	32	135	102	147	69	181	113	167	121	101	95
227	75	176	13	42	92	110	9	214	123	230	99	157	66	54	249
210	242	51	0	22	7	161	156	209	195	236	46	162	49	120	31
17	44	146	179	246	169	26	79	245	234	48	78	222	23	77	219
127	20	117	163	67	65	91	128	112	11	153	18	144	82	111	64
119	76	139	241	218	223	253	27	105	14	217	8	170	175	15	207
96	53	2	251	134	204	60	43	34	55	172	229	233	228	80	104
198	68	235	224	100	164	52	28	151	86	155	126	30	4	244	129
188	184	39	3	141	19	238	186	252	142	213	137	143	165	240	83
106	24	180	16	248	177	122	114	85	133	196	255	231	12	38	168
115	239	201	192	194	25	174	59	72	47	116	221	136	237	247	152
36	150	140	199	98	212	225	61	166	190	243	97	107	208	171	103
94	226	45	90	197	211	132	84	130	93	118	57	250	40	187	71

3.4 Correctness of the Proposed Scheme

We can verify the scheme proposed in Section 3.2 as follows:

- Since the proposed scheme works for even number of images, therefore we take 4 RGB images, as explained in Step 1 of the proposed scheme.
- After seperating the red, green, and blue components each image can be observed as,

  $I_1 = I_{R1}, I_{G1}, I_{B1}$

  $I_2 = I_{R2}, I_{G2}, I_{B2}$

  $I_3 = I_{R3}, I_{G3}, I_{B3}$

  $I_4 = I_{R4}, I_{G4}, I_{B4}.$
- For obtaining the $F_1, (U_{RK}, U_{GK}, U_{BK})$ can be taken as,

  $U_{RK} = (I_{R1} \oplus I_{R2} \oplus I_{R3} \oplus I_{R4})$

  $U_{GK} = (I_{G1} \oplus I_{G2} \oplus I_{G3} \oplus I_{G4})$

  $U_{BK} = (I_{B1} \oplus I_{B2} \oplus I_{B3} \oplus I_{B4}).$
- $F_2 = (U_{RN}, U_{GN}, U_{BN})$ can be computed as follows:

  $U_{RN} = (\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4})$

  $U_{GN} = (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4})$

  $U_{BN} = (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4}).$
- Same number of shares must be generated as that of images that is if we are using 4 images then there must be 4 Shares generated.

  $Share_1 = (I_{R1}, I_{G1}, I_{B1}) \oplus ((\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4}))$

  $\Rightarrow Share_1 = ((I_{R1} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G1} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B1} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})$

  $Share_2 = (I_{R2}, I_{G2}, I_{B2}) \oplus ((\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4}))$

  $\Rightarrow Share_2 = ((I_{R2} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G2} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B2} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})$

$Share_3 = (I_{R3}, I_{G3}, I_{B3}) \oplus ((\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4}))$

$\Rightarrow Share_3 = ((I_{R3} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G3} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B3} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})$

$Share_4 = (I_{R4}, I_{G4}, I_{B4}) \oplus ((\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4}))$

$\Rightarrow Share_4 = ((I_{R4} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G4} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B4} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})$

- Apply Boolean operation XOR on all the shares to regain F_1.

  $Share_1 \oplus Share_2 \oplus Share_3 \oplus Share_4 = \{(I_{R1} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G1} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B1} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\} \oplus \{(I_{R2} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G2} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B2} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\} \oplus \{(I_{R3} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G3} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B3} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\} \oplus \{(I_{R4} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G4} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B4} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\}$

  $\Rightarrow Share_1 \oplus Share_2 \oplus Share_3 \oplus Share_4 = \{(I_{R1} \oplus I_{R2} \oplus I_{R3} \oplus I_{R4}), (I_{G1} \oplus I_{G2} \oplus I_{G3} \oplus I_{G4}), (I_{B1} \oplus I_{B2} \oplus I_{B3} \oplus I_{B4})\}$

  $\Rightarrow \{(I_{R1} \oplus I_{R2} \oplus I_{R3} \oplus I_{R4}), (I_{G1} \oplus I_{G2} \oplus I_{G3} \oplus I_{G4}), (I_{B1} \oplus I_{B2} \oplus I_{B3} \oplus I_{B4})\} = F_1$

- After applying the bit complement operation on F_1, F_2 will be of the form,

  $\{(\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\} = F_2$

- Images will be recovered if XOR operation is implemented on each *Share* and F_2.

  $Share_1 \oplus F_2 = \{(I_{R1} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G1} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B1} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\} \oplus \{(\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\}$

  $\Rightarrow Share_1 \oplus F_2 = (I_{R1}, I_{G1}, I_{B1})$

  $\Rightarrow (I_{R1}, I_{G1}, I_{B1}) = I_1$

  $Share_2 \oplus F_2 = \{(I_{R2} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G2} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B2} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\} \oplus \{(\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\}$

$\Rightarrow Share_2 \oplus F_2 = (I_{R2}, I_{G2}, I_{B2})$

$\Rightarrow (I_{R2}, I_{G2}, I_{B2}) = I_2$

$Share_3 \oplus F_2 = \{(I_{R3} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G3} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B3} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\} \oplus \{(\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\}$

$\Rightarrow Share_3 \oplus F_2 = (I_{R3}, I_{G3}, I_{B3})$

$\Rightarrow (I_{R3}, I_{G3}, I_{B3}) = I_3$

$Share_4 \oplus F_2 = \{(I_{R4} \oplus \sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (I_{G4} \oplus \sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (I_{B4} \oplus \sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\} \oplus \{(\sim I_{R1} \oplus \sim I_{R2} \oplus \sim I_{R3} \oplus \sim I_{R4}), (\sim I_{G1} \oplus \sim I_{G2} \oplus \sim I_{G3} \oplus \sim I_{G4}), (\sim I_{B1} \oplus \sim I_{B2} \oplus \sim I_{B3} \oplus \sim I_{B4})\}$

$\Rightarrow Share_4 \oplus F_2 = (I_{R4}, I_{G4}, I_{B4})$

$\Rightarrow (I_{R4}, I_{G4}, I_{B4}) = I_4$

3.5 Example

Image Encryption

In this example we will encrypt four different RGB images by using boolean XOR operation and S-Box.

- **Step 1**

Initially, we used "MATLAB" to read four different RGB images, as shown in the Figure 3.5

Image 1 = U_1

Image 2 = U_2

Image 3 = U_3

Image 4 = U_4

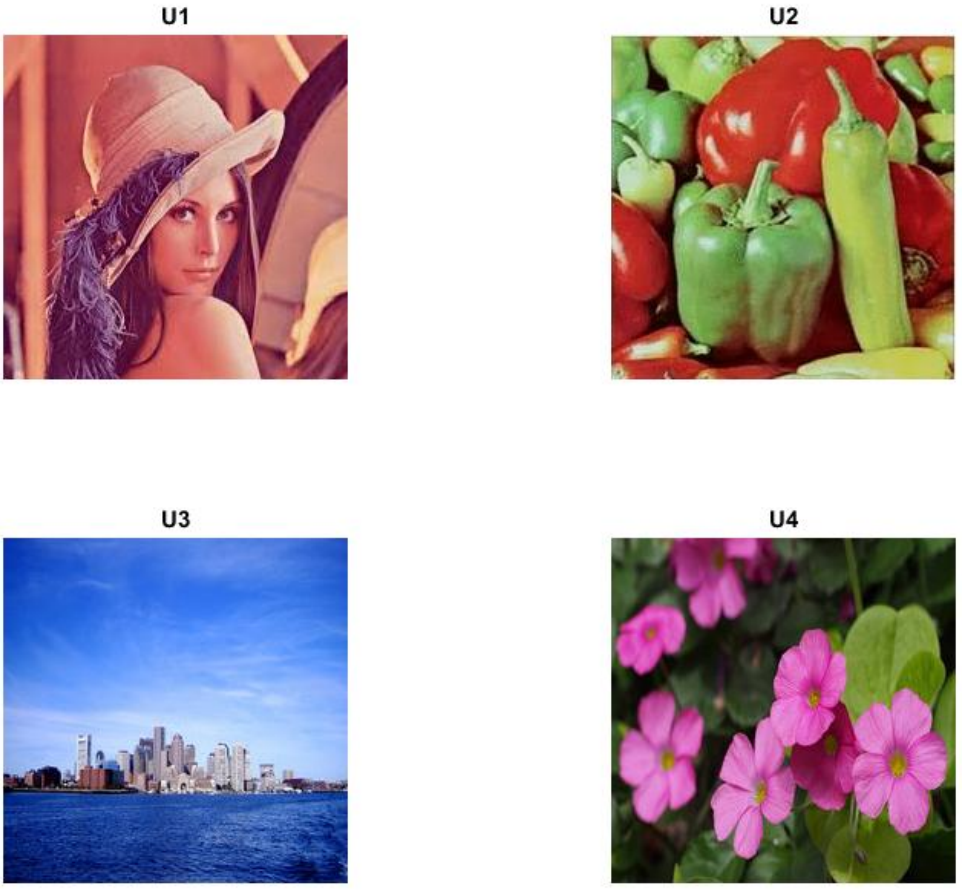

Figure 3.5: Original Images

- **Step 2**

Now, we separate red, green and blue color components of all the four images as,

$U_1 = (U_{r_1}, U_{g_1}, U_{b_1})$

$U_2 = (U_{r_2}, U_{g_2}, U_{b_2})$

$U_3 = (U_{r_3}, U_{g_3}, U_{b_3})$

$U_4 = (U_{r_4}, U_{g_4}, U_{b_4})$

- **Step 3**

After separating color components of each image, we compute bit XOR of red, green and blue colors of K images as shown in the Figure 3.6, for $K = 1, 2, 3, 4$.

$U_{r_K} = (U_{r_1} \oplus U_{r_2} \oplus U_{r_3} \oplus U_{r_4})$

$U_{g_K} = (U_{g_1} \oplus U_{g_2} \oplus U_{g_3} \oplus U_{g_4})$

$U_{b_K} = (U_{b_1} \oplus U_{b_2} \oplus U_{b_3} \oplus U_{b_4})$

$F_1 = (U_{r_K}, U_{g_K}, U_{b_K})$

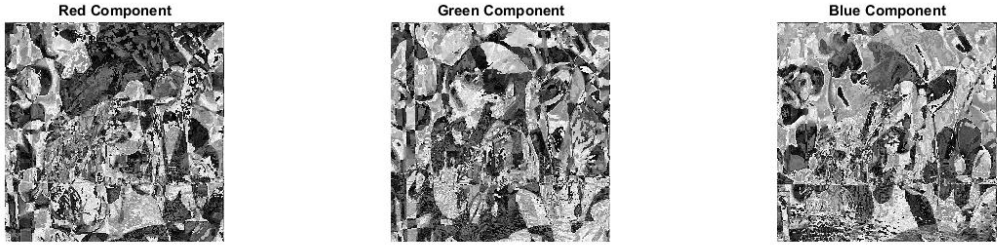

Figure 3.6: Bit XOR

- **Step 4**

We have evaluated bit-wise complement of $U_{r_K}, U_{g_K}, U_{b_K}$ seperately *i.e.*

$U_{r_N} = \sim U_{r_K}$

$U_{g_N} = \sim U_{g_K}$

$U_{b_N} = \sim U_{b_K}$

$F_2 = (U_{r_N}, U_{g_N}, U_{b_N})$

- **Step 5**

In this step we create shares of all the images shown in Figure 3.7 taken in Step 1.

$Share_1 = (I_1 \oplus F_2)$

$Share_2 = (I_2 \oplus F_2)$

$Share_3 = (I_3 \oplus F_2)$

$Share_4 = (I_4 \oplus F_2)$

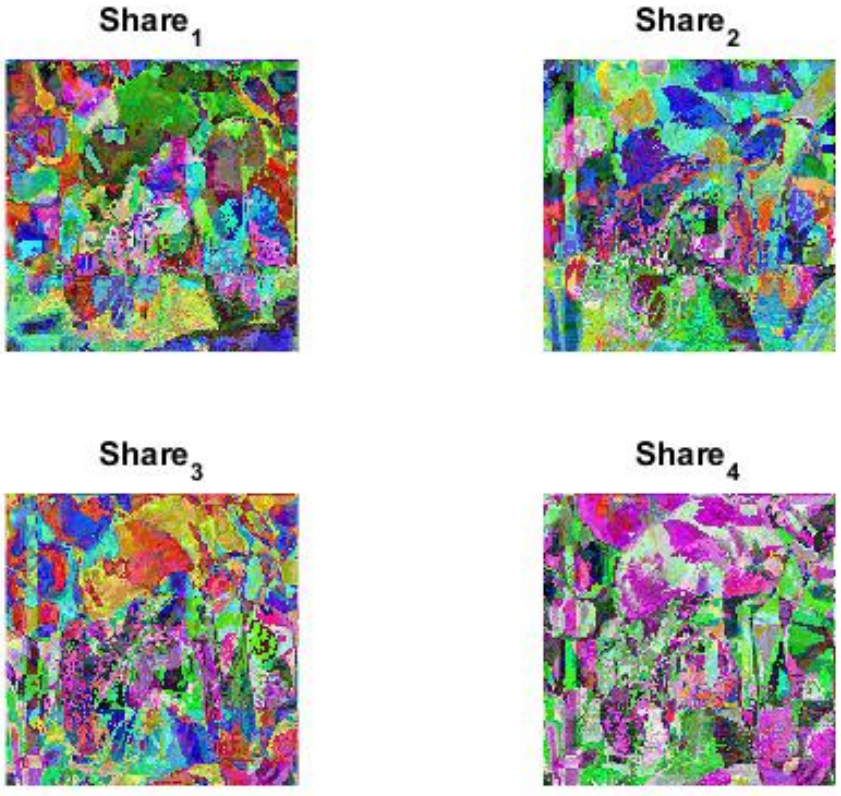

Figure 3.7: Shares of Secret Images

- **Step 6**

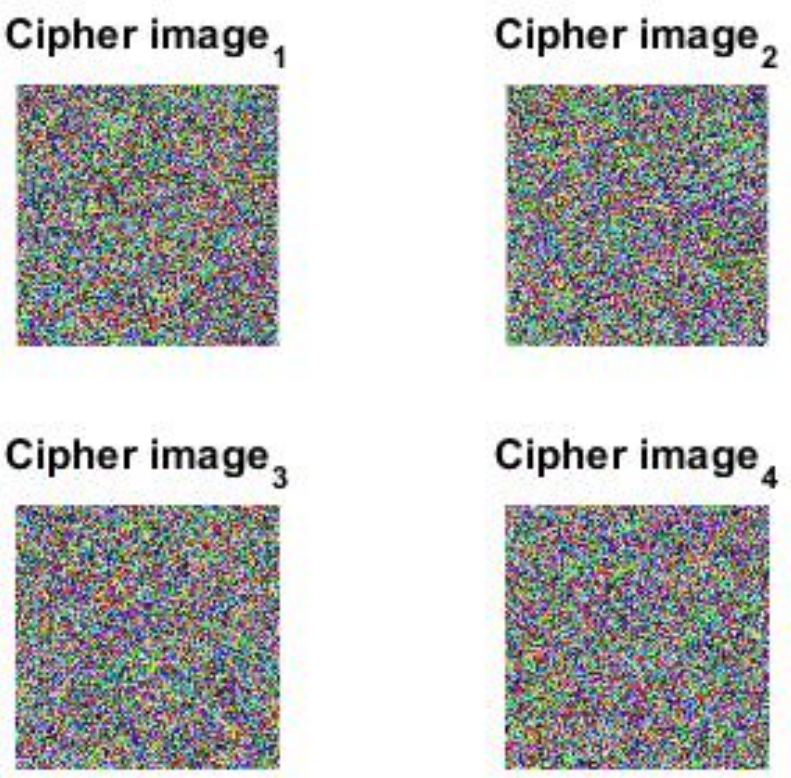

Figure 3.8: Cipher Images

After constructing shares of original images, we apply S-Box on each share individually and get the cipher images as shown in the Figure 3.8.

Cipher $image_1 = \mathrm{S}(Share_1)$

Cipher $image_2 = \mathrm{S}(Share_2)$

Cipher $image_3 = \mathrm{S}(Share_3)$

Cipher $image_4 = \mathrm{S}(Share_4)$

Decryption

- **Step 1**

To recover the shares, we apply inverse S-Box *i.e* S^{-1} on the cipher images. We can observe the recovered shares in Figure 3.9.

$Share_1 = S^{-1}$ (Cipher $image_1$)

$Share_2 = S^{-1}$ (Cipher $image_2$)

$Share_3 = S^{-1}$ (Cipher $image_3$)

$Share_4 = S^{-1}$ (Cipher $image_4$)

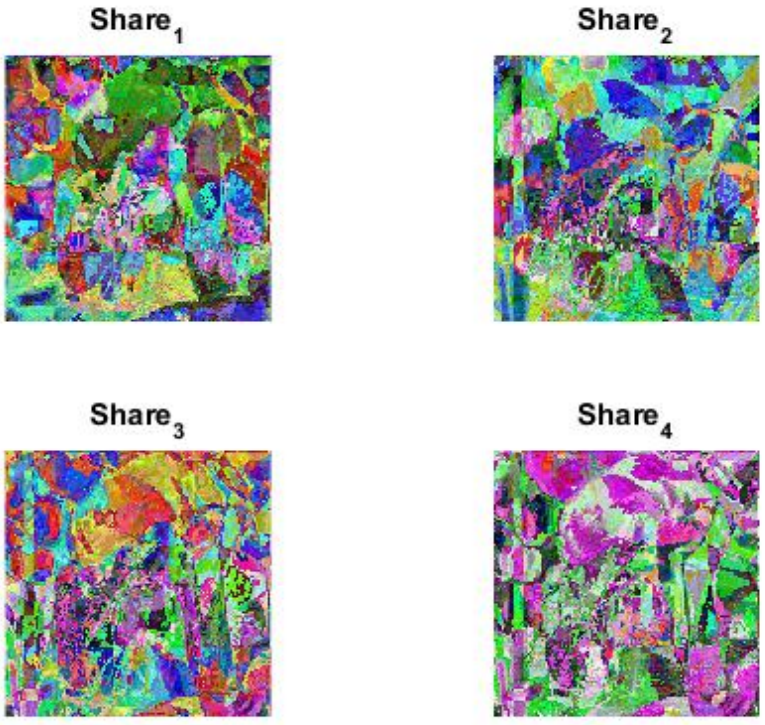

Figure 3.9: *Recovered Shares*

- **Step 2**

We apply XOR operation on all the shares obtained in Step 7. We can see the XOR on all shares in Figure **??**.

$F_1 = Share_1 \oplus Share_2 \oplus Share_3 \oplus Share_4$

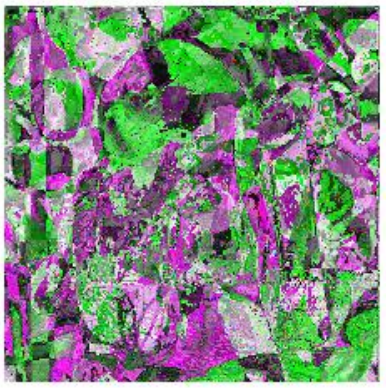

Figure 3.10: XOR of all *Shares*

- **Step 3**

We computed bit complement on F_1

that is, $F_2 = \sim F_1$.

- **Step 4**

For acquiring the plain images from shares we apply XOR on each share of Step 8 and F_2 of Step 9 as shown in the Figure 3.11.

$U_1 = Share_1 \oplus F_2$

$U_2 = Share_2 \oplus F_2$

$U_3 = Share_3 \oplus F_2$

$U_4 = Share_4 \oplus F_2$.

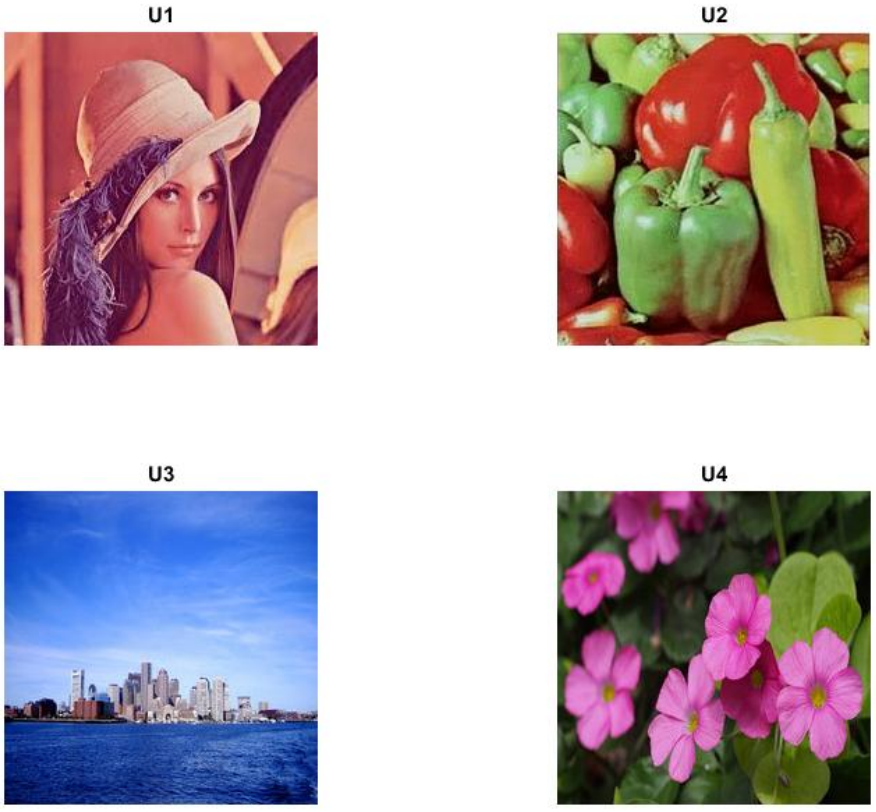

Figure 3.11: Recovered Original Images

Chapter 4

Results and Discussion

In this chapter, we will discuss the results of the proposed scheme (explained in Chapter 3) with the assistance of several approaches. For experimental verification of the proposed scheme we used windows 10 pro operating system, core $i5$ and 4 GB RAM using the "MATLAB (R2015a)" software. We also analyze our scheme using histogram test, correlation and entropy test. Security of the scheme can be analyzed in Section 4.3.

4.1 Analysis of Proposed S-Box

We have examined some desirable properties of proposed S-Box by a "MATLAB Tool" for evaluation of S-Box. The results are displayed as follows:

1. Balanced = Yes
2. Bijective = Yes
3. Hamming Distance of all boolean functions.
4. Each column has Hamming weight

$$[128 \quad 128 \quad 128 \quad 128 \quad 128 \quad 128 \quad 128 \quad 128]$$

5. Fixed Point = 0.
6. Opposite Fixed Point = 2.
7. Almost Bent Non-linearity = 116.6868
8. Maximum Non-linearity = 120
9. Non-linearity of S-Box = 104
10. Differential Branch number = 3
11. Dynamic Distance of Boolean function

$$[12 \quad 10 \quad 10 \quad 12 \quad 14 \quad 12 \quad 12 \quad 12]$$

12. Number of functions satisfying Avalanche criteria = 39.
13. S-Box function strict Avalanche values

$$[1080 \quad 1024 \quad 1032 \quad 1004 \quad 1012 \quad 1080 \quad 1024 \quad 1044]$$

14. Percentage Avalanche effects in S-Box = 60.9375

4.2 Experimental Results

This section consists of the experimental results on proposed "**Secure and Efficient Multi Secret Image Sharing Scheme based on Boolean Operation XOR and S-Box Encryption**". For the experimental verification of our scheme, four RGB images named as Lena, Peppers, Building and Flowers are taken. Results of the proposed scheme are shown in the Figure 4.1.

4.2.1 Performance Analysis

We checked the resistance to several attacks of the suggested scheme after applying some popular security tests. For this purpose, images of Lena, Peppers, Building and Flower of same size (225×225) are used.

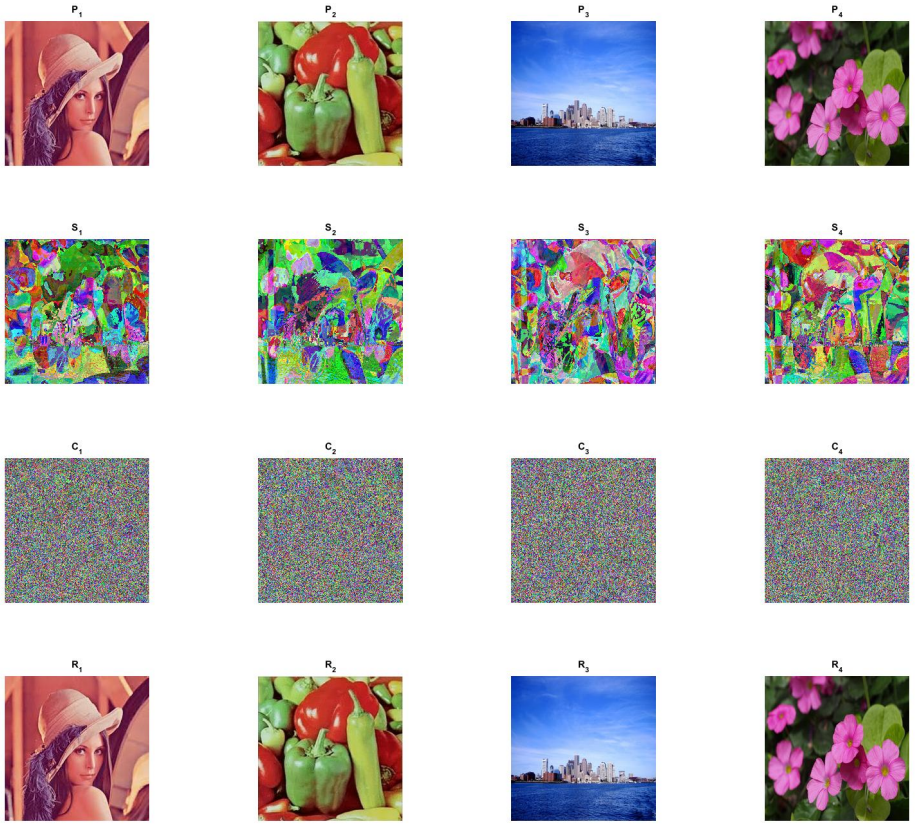

Figure 4.1: Results of Whole Scheme

1. **Lossless Encryption**

   The decoding process shows that the discussed scheme is lossless.

2. **Statistical Attack**

A secure cryptosystem must resist all type of statistical attacks efficiently. For examining the resistance of proposed cryptosystem against statistical attacks, we use the histogram test, correlation, and entropy test.

- **Histogram Test**

The pictorial representation of each pixel intensity value and their frequencies is known as histogram [48]. It describes the confusion and diffusion properties in the encrypted images. We can conclude that the proposed scheme satisfies the histogram test after comparing the histogram of plain images in Figure 4.2 and

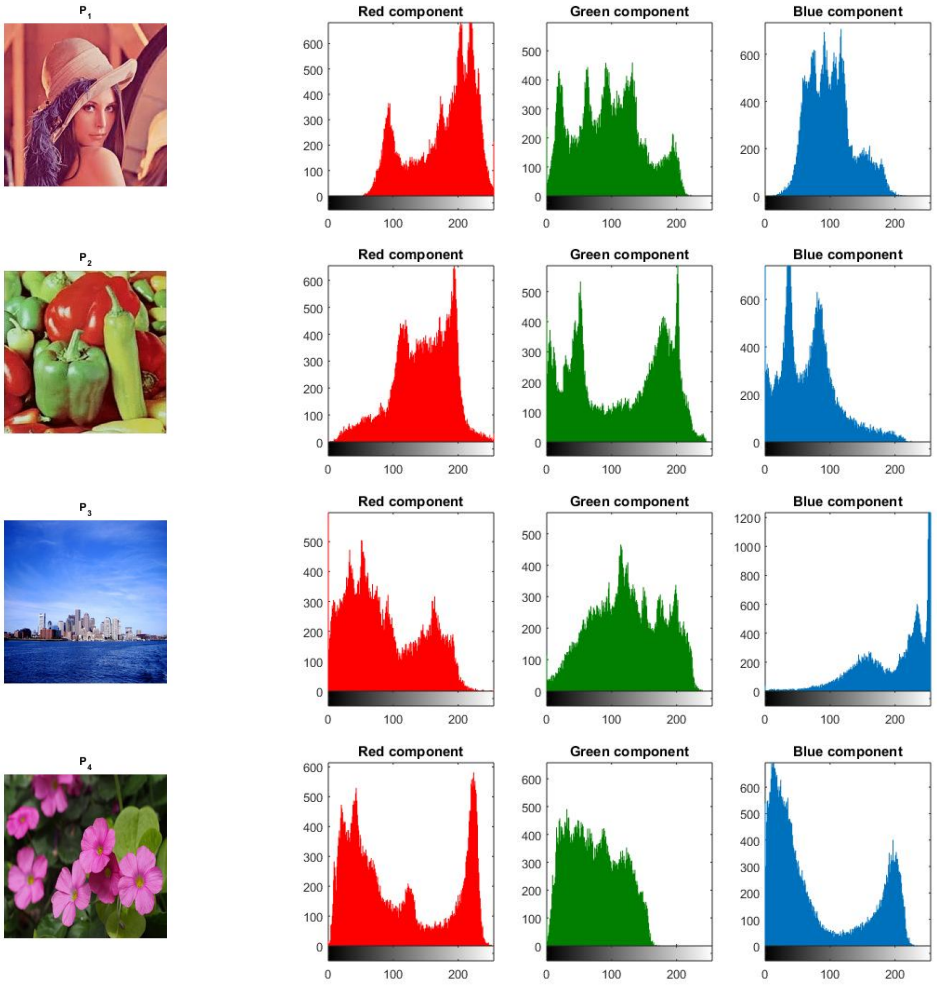

Figure 4.2: Histogram of Original Images

the histogram of cipher images in Figure 4.3. Since the histogram of cipher images is almost uniform, therefore an attacker can not breaks the security of the proposed scheme.

- **Correlation Test**

Correlation [49] means the relation of neighbouring pixels in horizontal, vertical and diagonal directions. In plain images, pixels are highly correlated with each other. In cipher images there is no or a slight correlation among the adjacent pixels. That's why a cryptanalyst cannot get any information about the plain image.

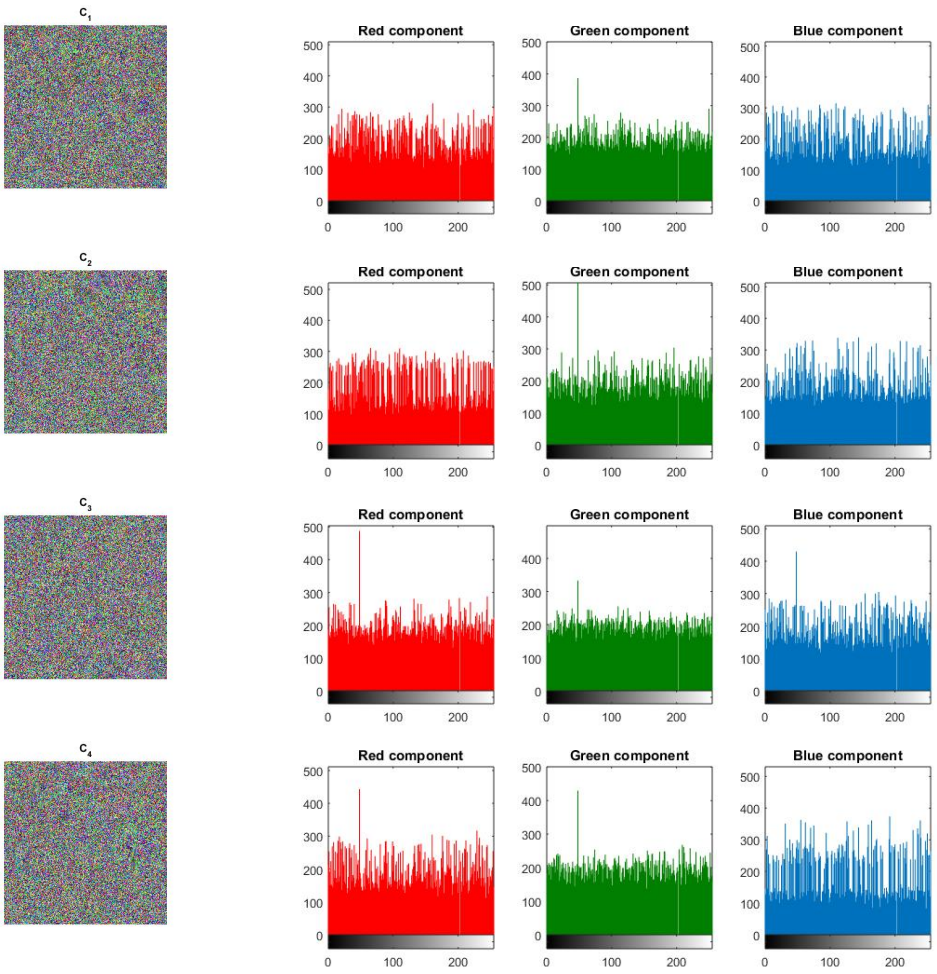

Figure 4.3: Histogram of Cipher Images

Table 4.1: **Correlation of Original Images**

Images	**Row Correlation**	**Column Correlation**	**Diagonal Correlation**
Image1	0.9527	0.9107	0.8794
Image2	0.9579	0.9452	0.9150
Image3	0.9603	0.9729	0.9470
Image4	0.9845	0.9728	0.9599

We use the Equation (4.1) for finding the correlation in horizontal, vertical and diagonal pixels of original and cipher images where, s and t are the values of two neighbouring pixels in original and cipher images. N is the total number of

adjacent pixels in the choosen image. "MATLAB" command $corr2(s,t)$ is used to check the correlation the adjacent pixels.

$$r_{st} = \frac{N\sum_{i=1}^{N}(s_i \times t_i) - (\sum_{i=1}^{N} s_i)(\sum_{i=1}^{N} t_i)}{\sqrt{(N\sum_{i=1}^{N}(s_i)^2 - (\sum_{i=1}^{N} s_i)^2) \times (N\sum_{i=1}^{N}(t_i)^2 - (\sum_{i=1}^{N} t_i)^2)}} \tag{4.1}$$

Table 4.2: **Correlation of Cipher Images**

Images	**Row Correlation**	**Column Correlation**	**Diagonal Correlation**
Image1	0.0191	0.0127	0.0113
Image2	0.0211	0.0183	0.0082
Image3	0.0149	0.0111	0.0070
Image4	0.0185	0.0116	0.0074

From Table 4.1 we can examine the correlation of original images and Table 4.2 shows the correlation between cipher images. Images are said to be linearly correlated if the value of the correlation test is close to 1 [50, 51]. If the value of correlation is close to 0 then there is no correlation in the images. After analyzing Table 4.2 we can easily conclude that the proposed encoding process satisfies the correlation test since, the adjacent pixels of cipher images are very close to zero.

- **Entropy Test**

We apply entropy [52] test for accurately computing randomness in the behavior of a data set. It is an essential and systematic test for checking whether the pixels of the cipher image are random or not. We can calculate entropy [53] E of T that is $E(T)$ by the following formula:

$$E(T) = -\sum_{i=0}^{n} p(t_i)\, log_2\,(p(t_i)) \tag{4.2}$$

where, n = number of different values of data and $p(t_i)$ is the occurence probability of t_i values. A data set with entropy nearest to it's ideal value is 8 is said to be random. We executed the entropy test on cipher images, Table 4.3 shows entropy results for four cipher images.

Table 4.3: **Difference of Entropy between Original Images and Cipher Images**

Images	**Original Images**	**Cipher Images**
Image1	7.3172	7.9572
Image2	7.4390	7.9553
Image3	7.3354	7.9796
Image4	7.3922	7.9654

It is clear from the results of Table 4.3 that entropy value of all cipher images are close to 8, hence the suggested cryptosystem is suitable for creating high randomness in cipher images.

4.3 Key Management

A collection of skills and methodologies to aid the formation and continuation of keying relationship between the legitimate users is known as key management. For the assistance of cryptographic techniques, conversational parties develop an association of keys. This association works to share common secret data. The foundations of cryptographic security rely on the pillars of key management. It gives data confidentiality and authenticity of the communicating parties and data as well. It also provides data integrity in digital signatures. An objective of crypto techniques is to truncate complicated problems and to secure the keys with efficiency.

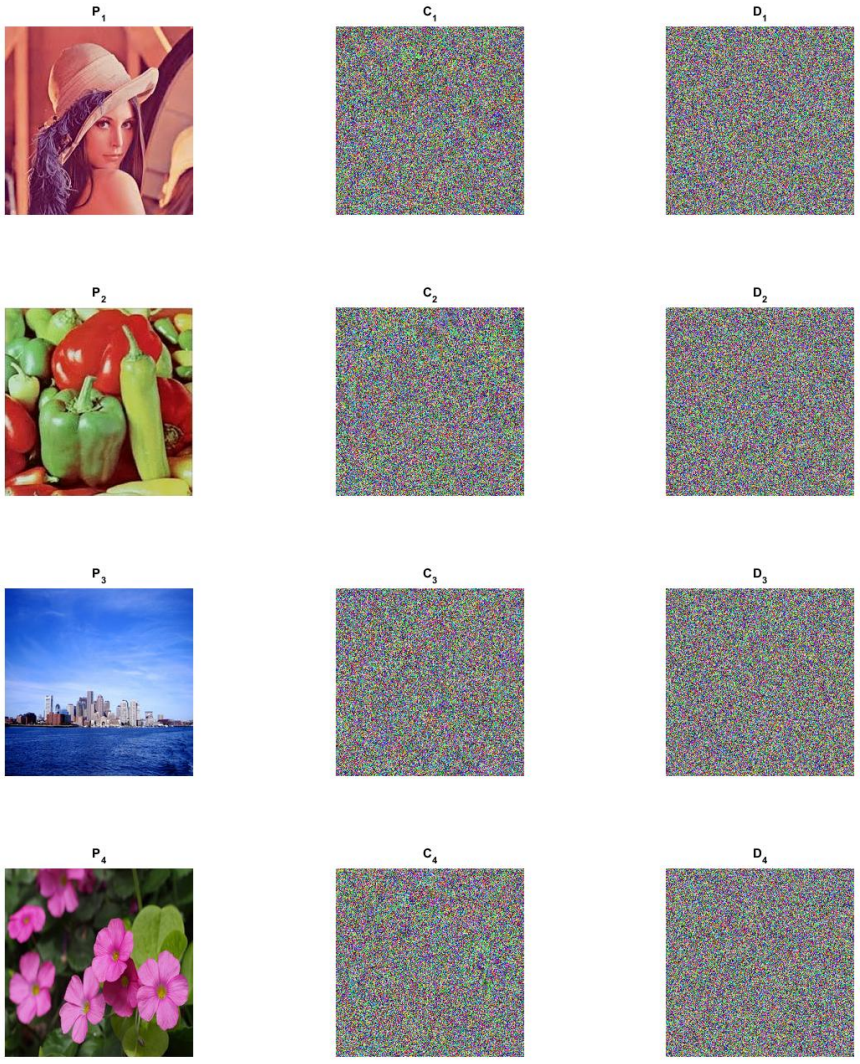

Figure 4.4: Key Sensitivity Analysis

- **Key Analysis**

There are many attacks for breaking a cryptosystem. Large key size is essential for a secure cryptosystem. Key space analysis [54] and key sensitivity are used for key evaluation. A brief description of these tests and their experimental results are given below.

1. Key Space Analysis

Key space means the total number of different keys used in encoding. Key spacing satisfies the resistance of a cryptosystem against the brute force attack. Generally,

a cryptosystem is said to be secure if it uses a key size of more than 160 bits. In the proposed security technique, the parameters α, β, $K(a_1, a_2)$ are used as secret keys. Each of these secret keys must be of the size 2^{160}. Hence the resistance of suggested cryptosystem against the brute force attack is remarkable.

2. Analysis of Key Sensitivity

The sensitivity of a key means that a minor change in the keys would generate remarkably different cipher images. So, a secure cryptosystem should be highly sensitive to keys. Suppose a key K_1 is obtained by changing one bit of the original key K. Then by the sensitivity of keys, we mean that the images recovered by the changed key K_1 will not reveal any information about the original images. For instance, if $K = (1355, 2421)$ and by changing one bit in K we have $K_1 = (1387, 2389)$. In Figure 4.4 P_1, P_2, P_3 and P_4 are the original images, C_1, C_2, C_3 and C_4 are cipher images encrypted by key K and D_1, D_2, D_3 and D_4 are decrypted images by using key K_1. Thus D_i in Figure 4.4 idicates that the original images are not revealed at decryption stage by using K_1. Hence the proposed scheme is sensitive to key and highly secure.

4.4 Conclusion

From the above results, it is concluded that the proposed methodology is secure and efficient. Processing images with ECC encryption "MATLAB" takes more time whereas the same task is completed efficiently by using the S-Box encryption. Therefore, S-Box encryption reduces the computational cost in comparison with ECC encryption. The use of S-Box based on ECC points is used to make the shares more secure from the intruders. Usually, it is really very hard to sustain the shares information while working with secret image sharing. However, we resolved this issue by using a two-way encryption one is the share generation and other is the mapping of share's pixels to S-box values. Moreover, the known plaintext and chosen plaintext attacks do not harm the security of the proposed scheme since

the two-way encryption (share generation and S-Box encryption) is applied in the suggested scheme. Results of the scheme exhibit that it is highly secure, sensitive and has less computational cost.

Bibliography

[1] A. Devi, A. Sharma, and A. Rangra, "A review on des, aes and blowfish for image encryption and decryption," *International Journal of Computer Science and Information Technologies*, vol. 6, no. 3, pp. 3034–3036, 2015.

[2] A. Verma, P. Guha, and S. Mishra, "Comparative study of different cryptographic algorithms," *International Journal of Emerging Trends and Technology in Computer Science*, vol. 5, no. 2, pp. 58–63, 2016.

[3] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure password-authenticated key exchange using diffie-hellman," pp. 156–171, 2000.

[4] T. Takagi, "Fast rsa-type cryptosystem modulo p k q," pp. 318–326, 1998.

[5] R. Singh and S. Kumar, "Elgamals algorithm in cryptography," *International Journal of Scientific and Engineering Research*, vol. 3, no. 12, pp. 1–4, 2012.

[6] J. H. Silverman, "An introduction to the theory of lattices and applications to cryptography," *Computational Number Theory and Applications to Cryptography, University of Wyoming*, pp. 1–212, 2006.

[7] A. K. Lenstra, "Integer factoring," pp. 31–58, 2000.

[8] I. A. Semaev, "Summation polynomials and the discrete logarithm problem on elliptic curves." *IACR Cryptology ePrint Archive*, vol. 2004, pp. 1–31, 2004.

[9] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[10] G. R. Blakley, "Safeguarding cryptographic keys," vol. 48, pp. 313–317, 1979.

[11] T. H. Chen and C. S. Wu, "Efficient multi-secret image sharing based on boolean operations," *Signal Processing*, vol. 91, no. 1, pp. 90–97, 2011.

[12] C. C. Chen and W. J. Wu, "A secure boolean-based multi-secret image sharing scheme," *Journal of Systems and Software*, vol. 92, pp. 107–114, 2014.

[13] K. Shankar and P. Eswaran, "A secure visual secret share (vss) creation scheme in visual cryptography using elliptic curve cryptography with optimization technique," *Australian Journal of Basic and Applied Science*, vol. 9, no. 36, pp. 150–163, 2015.

[14] K. Shankar, G. Devika, and M. Ilayaraja, "Secure and efficient multi-secret image sharing scheme based on boolean operations and elliptic curve cryptography," *International Journal of Pure and Applied Mathematics*, vol. 116, no. 10, pp. 293–300, 2017.

[15] U. Hayat and N. A. Azam, "A novel image encryption scheme based on an elliptic curve," *Signal Processing*, vol. 155, pp. 391–402, 2019.

[16] K. Jacobs, "A survey of modern mathematical cryptology," pp. 1–13, 2011.

[17] R. Pakshwar, V. K. Trivedi, and V. Richhariya, "A survey on different image encryption and decryption techniques," vol. 4, pp. 113–116, 2013.

[18] MATLAB. Matlab help. [Online]. Available: https://www.mathworks.com/help

[19] K. Ruohonen, "Mathematical cryptology," *Lecture Notes*, pp. 1–138, 2010.

[20] W. Stallings, "Cryptography and network security: principles and practices," pp. 1–457, 2006.

[21] A. J. Menezes, "Handbook of applied cryptography," pp. 1–794, 1997.

[22] A. Sangeeta, Kaur, "A review on symmetric key cryptography algorithms," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 4, pp. 358–361, 2017.

[23] D. G. Amalarethinam and J. S. Geetha, "Image encryption and decryption in public key cryptography based on mr," pp. 133–138, 2015.

[24] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[25] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[26] M. Matsui, "Linear cryptanalysis method for des cipher," pp. 386–397, 1993.

[27] G. Bhatnagar and Q. J. Wu, "Selective image encryption based on pixels of interest and singular value decomposition," *Digital signal processing*, vol. 22, no. 4, pp. 648–663, 2012.

[28] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," pp. 349–356, 2009.

[29] A. McAndrew, "An introduction to digital image processing with matlab notes for scm2511 image processing," *School of Computer Science and Mathematics, Victoria University of Technology*, vol. 264, no. 1, pp. 1–264, 2004.

[30] R. Pakshwar, V. K. Trivedi, and V. Richhariya, "A survey on different image encryption and decryption techniques," *International journal of computer science and information technologies*, vol. 4, no. 1, pp. 113–116, 2013.

[31] R. Lidl and H. Niederreiter, "Introduction to finite fields and their applications," pp. 1–416, 1994.

[32] C. J. Benvenuto, "Galois field in cryptography," *University of Washington*, pp. 1–11, 2012.

[33] A. Beimel, "Secure schemes for secret sharing and key distribution," pp. 1–115, 1996.

[34] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 72, no. 9, pp. 56–64, 1989.

[35] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," pp. 148–164, 1999.

[36] R. Fuji Hara and Y. Miao, "Ideal secret sharing schemes: Yet another combinatorial characterization, certain access structures, and related geometric problems," pp. 1–16, 2018.

[37] O. Farràs, T. B. Hansen, T. Kaced, and C. Padró, "On the information ratio of non-perfect secret sharing schemes," *Algorithmica*, vol. 79, no. 4, pp. 987–1013, 2017.

[38] S. S. M. Jarecki, "Proactive secret sharing and public key cryptosystems," Ph.D. dissertation, Massachusetts Institute of Technology, 1995.

[39] M. Stadler, "Publicly verifiable secret sharing," pp. 190–199, 1996.

[40] M. R. Mudge, "Elliptic curves," *The Mathematical Gazette*, vol. 84, no. 500, pp. 364–364, 2000.

[41] E. Yin, "Curve selection in elliptic curve cryptography," *published by San Jose State University*, pp. 1–7, 2005.

[42] W. Zhang and E. Pasalic, "Highly nonlinear balanced s boxes with good differential properties," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7970–7979, 2014.

[43] L. R. Knudsen and M. Robshaw, "The block cipher companion," 2011.

[44] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact rijndael hardware architecture with s-box optimization," pp. 239–254, 2001.

[45] C. C. Thien and J. C. Lin, "Secret image sharing," *Computers and Graphics*, vol. 26, no. 5, pp. 765–770, 2002.

[46] P. Astya, B. Singh, and D. Chauhan, "Image encryption and decryption using elliptic curve cryptography," *International Journal of Advance Research In Science And Engineering IJARSE*, no. 3, 2014.

[47] S. K. Nerella, K. V. Gadi, and R. S. Chaganti, "Securing images using colour visual cryptography and wavelets," *International journal of advanced research in computer science and software engineering*, vol. 2, no. 3, 2012.

[48] G. R. S. Qaid, "Encryption and decryption image using multiobjective soft computing algorithm," pp. 1–19, 2015.

[49] S. Somaraj and M. A. Hussain, "Performance and security analysis for image encryption using key image," *Indian Journal of Science and Technology*, vol. 8, no. 35, 2015.

[50] Y. Wu, Y. Zhou, J. P. Noonan, and S. Agaian, "Design of image cipher using latin squares," *Information Sciences*, vol. 264, pp. 317–339, 2014.

[51] M. Ahmad and T. Ahmad, "A framework to protect patient digital medical imagery for secure telediagnosis," *Procedia engineering*, vol. 38, pp. 1055–1066, 2012.

[52] M. A. Al-Husainy and D. M. Uliyan, "Image encryption technique based on the entropy value of a random block," *Image*, vol. 8, no. 7, 2017.

[53] M. Ahmad and A. Chopra, "Chaotic dynamic s boxes based substitution approach for digital images," pp. 1–12, 2017.

[54] L. Li, A. A. A. El-Latif, and X. Niu, "Elliptic curve elgamal based homomorphic image encryption scheme for sharing secret images," *Signal Processing*, vol. 92, no. 4, pp. 1069–1078, 2012.