Palestine Polytechnic University
Deanship of Graduate Studies and Scientific Research
Master of informatics

Selective Colour Image Encryption Based on MSB and Sensitive Bits

Submitted by:

Issa Jacaman

Supervisor:

Dr. Mousa Farajalla

Thesis submitted in partial fulfilment of requirements of the
Master degree of Science in Informatics
September, 2023

The undersigned hereby certify that they have read, examined and recommended to the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University the approval of a thesis entitled: **Selective Colour Image Encryption Based on MSB and Sensitive Bits**, submitted by **Issa Jacaman** in partial fulfilment of the requirements for the degree of Master in Informatics.

Graduate Advisory Committee:

Dr. Mousa Farajallah (Supervisor), Palestine Polytechnic University.

Signature:____________________ Date:______________

Dr. Radwan Tahboub (Internal committee member), Palestine Polytechnic University.

Signature:____________________ Date:______________

Dr. Majdi Owda, (External committee member), Arab American University.

Signature:____________________ Date:______________

Thesis Approved

Dr. Nafeth Nasereddin
Dean of Graduate Studies and Scientific Research
Palestine Polytechnic University

Signature:____________________ Date:______________

DECLARATION

I declare that the Master Thesis entitled **"Selective Colour Image Encryption Based on MSB and Sensitive Bits"** is my original work, and hereby certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgement is made in the text.

Issa Jacaman

Signature:____________________ Date:________________

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for the master degree in Informatics at Palestine Polytechnic University, I agree that the library shall make it available to borrowers under rules of the library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from, reproduction, or publication of this thesis may be granted by my main supervisor, or in his absence, by the Dean of Graduate Studies and Scientific Research when, in the opinion of either, the proposed use of the material is for scholarly purposes.

Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Issa Jacaman

Signature:____________________ Date:_______________

الملخص

من أحد أهم التطورات التكنولوجية في هذا العقد، استخدام الأجهزة الإلكترونية ذات المصادر المحدودة مثل إنترنت الأشياء. تعالج هذه الأجهزة كميات هائلة من الصور باستخدام المصادر المتاحة لها من الذاكرة والمعالج. لذلك، يتطلب وجود خوارزميات تقوم بتشفير الصور الملونة مع مراعاة القدرات الإلكترونية المحدودة لهذه الأجهزة. يجدر بالذكر أن خوارزميات التشفير التقليدية غير مناسبة لتشفير الصور بسبب الترابط والتشابه في قيم البكسل داخل الصورة نفسها. وبسبب هذه الخاصية للصور، فإن الخوارزميات التقليدية لا تراعي محدودية المصادر لهذه الأجهزة.

تهدف هذه الرسالة إلى اقتراح خوارزمية تقوم بتشفير الصور الملونة، والتي تختار فقط الأجزاء المهمة داخل الصورة لعملية التشفير، وبذلك تقوم بتقليل المصادر المطلوبة لتشفير الصور. قمنا بإجراء اختبارات أداء لهذه الخوارزمية المقترحة في كلا من حالتي: التشفير الجزئي والكامل للصور الملونة. وبذلك، يمكننا فهم أكثر عن مدى كفاءة فعالية الخوارزمية المقترحة. من نتائج التقييم، حقق الخوارزمية التي اقترحناها قيمًا قريبة من المثلى للمقاييس التقييمية التي تم تنفيذها (المتوسط لـ $PSNR$ ٧٫٨، ٠٧٫٠ لـ $MSSIM$ و ٨٫٧ لـ IE)، مما يشير إلى نظام تشفير قوي. يمكن تحسين الخوارزمية لتشمل الصور ذات الدرجات الرمادية، وتطوير مولد أرقام عشوائية.

Abstract

Efficient encryption algorithms for coloured images are crucial for maintaining device performance when working with limited resources. Traditional algorithms can be too demanding and ineffective for safeguarding data on such devices. Resource-limited devices, such as the Internet of Things, actively collect images and send them through the public internet. With the increased use of such devices, more secure, robust encryption algorithms need to be developed. To overcome these issues, this thesis proposes an algorithm that encrypts coloured images. This algorithm only selects the important parts within the image for the encryption process, reducing the resources required to encrypt images. In addition to selective encryption, we also implemented the proposed algorithm for full image encryption. This allowed us to perform evaluation metrics over selective and full encryption, gaining insights into the differences between performing full and selective encryption. From the evaluation results, our proposed algorithm achieved near-optimal values (average PSNR 8.7, 0.07 for MSSIM and 7.8 for IE) for the performed evaluation metrics, indicating a robust cryptosystem. Our proposed algorithm can be enhanced to include grey-scale images and develop a pseudo-random number generator.

DEDICATION

I dedicate this thesis to my parents, my source of inspiration. Their belief and selflessness have been my main sources of support and resilience.

ACKNOWLEDGEMENT

I want to express my deep appreciation to my advisor, Dr Mousa Farajallah, for his unwavering support throughout my studies and research endeavours. His patience, encouragement, and extensive knowledge were instrumental in guiding me through researching and writing this thesis. I am truly grateful for his assistance and guidance every step of the way.

I want to thank the rest of my thesis committee, Dr Radwan Tahboub and Dr Majdi Owda, for their insightful comments and encouragement.

Last but not least, I want to thank my family, my parents and my siblings, for supporting me spiritually throughout writing this thesis. Moreover, I could not have completed this thesis without the support of my friends and colleagues.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ROI	Region Of Interest
MSB	Most Significant Bit
IV	Initial Vector
M	Most significant bit: exclusive-OR is executed on the MSB bit.
MC	In addition to the M process, the other three most significant bits are swapped among the RGB channel.
MCP	In addition to the MC process, pixels are scrambled within each block column-wise using the Fisher-Yates algorithm.
MCPB	In addition to the MCP process, an exclusive-OR is executed on all bytes
CBC	Cipher Block Chaining

Chapter 1

Introduction

Efficient encryption algorithms are crucial for secure data sharing on devices with limited resources. These algorithms need to balance energy and memory usage to optimise device performance. However, traditional algorithms are computationally intensive and can cause high communication overhead, which is ineffective for safeguarding data on limited resource devices. Encrypting image pixels presents a unique challenge because adjacent pixels in an image have similar values. To overcome these challenges, we have proposed an algorithm that achieves a strong cryptosystem. Our proposed solution is based on encrypting sensitive bits of a colour image's selected region of interest. The selection of a region of interest is based on an edge detection technique.

1.1 Problem Definition

As the age of big data emerges, an increasing amount of information is being digitised. While this brings convenience to people, it also raises concerns about personal privacy breaches and illegal data theft. Therefore, keeping digital images secure has become a new area of research.

Digital images contain vast data, including redundant information and high pixel correlation [18]. Image data possesses distinct and inherent qualities that distinguish it from traditional textual information. These qualities include a closely linked relationship between adjacent pixels and a high level of data redundancy. To accommodate these specific features, novel algorithms must be developed that differ from conventional cypher algorithms that are designed solely for textual data, such as Data Encryption Standard (DES) and Advanced Encryption Standard (AES) [25]. A proposal has been made for selective encryption over sensitive bits with the Fisher-Yates algorithm-based colour image encryption method to achieve a highly secure and efficient image encryption scheme.

1.2 Objectives

In this thesis, we aim to identify and pursue the following objectives:

- Develop a selective colour image encryption based on sensitive data targeting the region of interest (ROI).
- Evaluate each step of our encryption algorithm
- Adapt Fisher-Yates algorithm within our proposed algorithm.
- Define a proper methodology for selecting ROI.
- Adapt the same selective algorithm for encrypting the whole image.
- Get insights into the robustness of selective encryption versus full encryption for the same developed algorithm.
- Apply evaluation metrics over the proposed encryption algorithm (Selective and Full encryption).

- Compare the results of our evaluation metrics to related work.

1.3 Research Questions

The questions we aimed to address in this thesis are as follows:

- Is selective encryption considerable over full encryption?
- What are the best selection criteria of the region of interest of a colour plaintext image?
- How strong is the proposed encryption algorithm considering partially encrypting an image?

1.4 Contributions

In this thesis, an encryption algorithm for colour images is proposed. To selectively encrypt image blocks, the proposed algorithm employs edge detection criteria. These blocks contain important data about the image. The algorithm starts by performing an exclusive OR operation on the most significant bits of each pixel, followed by swapping the other three most significant bits between the pixel's three colour channels. The pixels are then scrambled vertically. Finally, all bytes are XOred according to a specific equation for a higher degree of confusion. Our proposed encryption algorithm adopted the Cipher Block Chaining mode of operation.

1.5 Thesis Organization

The remaining parts of the thesis are organized as follows: Chapter 2 describes the theories and basic concepts needed to understand the rest of the thesis, and it contains a summary of some previous works related to our work. Chapter 3 covers the methodology used in this thesis to enhance the accuracy. Chapter 4 demonstrates experiments and results achieved by the work and discusses the results. Finally, Chapter 6 concludes the work and proposes new directions for future work.

Chapter 2

Background and Related Work

2.1 Selective Encryption

The significance of selective image encryption cannot be overstated, as it allows for substantial savings in computations, cost, and time. Numerous efforts have been made in this regard, as conventional encryption algorithms for entire images may prove too immense. The authors of [20] stated that image data statistics vary significantly compared to traditional text data since these data are strongly correlated and have robust spatial/temporal redundancy. Selective encryption is an effective strategy because it allows for reliable security measures and computational requirements without any compromises [10]. There are several domains of encryption: spatial, frequency and hybrid [10].

Depending on the chosen criteria, selective encryption can be applied in either the frequency or spatial domain. In the frequency domain, specific frequency coefficients of image data are encrypted. However, selective encryption is done on the pixel or bit level in the spatial domain by confusing and diffusing their actual values [12].

2.2 Literature Review

Users share images and videos very frequently using their own resource-limited devices. This has brought the challenge of protecting private data while storing and transmitting data through the public internet with limited resource devices. Selective encryption [7] is considered the best solution for image encryption [8] in real-time resource-limited applications. Many researchers have adopted selective encryption to encrypt the important data of an image. Various methods were presented to enhance selective encryption, such as using the least significant bits and encryption within the MQ coding system (a context-based adaptive arithmetic coder). Regarding the previous review works, in the image encryption area, a few review works were done, such as [9] [14], they describe and evaluate (cryptoanalysis) the encryption algorithms concerning the image being in spatial, transform, spatiotemporal, optical, or compressive sensing domain. They consider encryption algorithms in a general overview of different domains. Meanwhile, in this brief review, we focus on the algorithms in the spatial domain. Furthermore, we categorized them following their encryption algorithm techniques.

2.2.1 Encryption Algorithms Categories

Encryption algorithms in [14] [26] [23] [21] [24] [16] can be categorized based on the selectivity method used in encrypting digital images. Figure 2.1 illustrates these categories; [21] used the In-compression approach, [14] calculated the coefficient correlation of pixel blocks, [24] considered encrypting only the most significant bits of an Image, [16] [26] encrypted only the edges using Edge detection, and [21] encrypted an image through its byte-stream with encryption ratio.

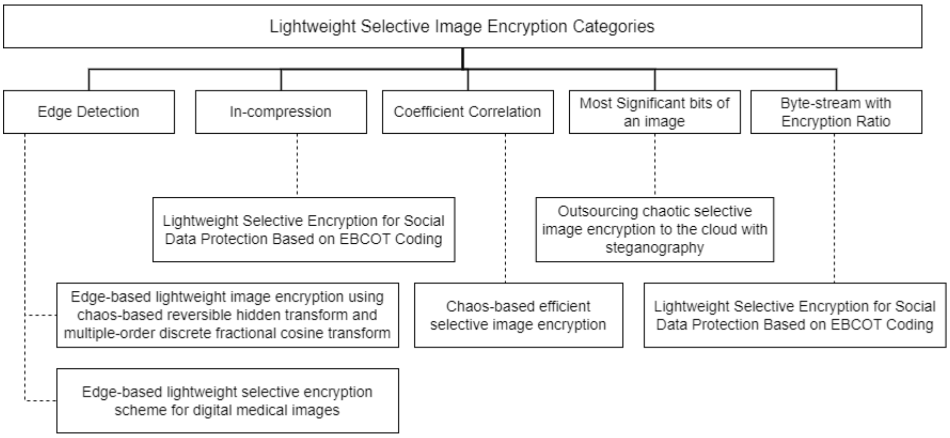

Lightweight Selective Image Encryption Categories
Edge Detection
In-compression
Coefficient Correlation
Most Significant bits of an image
Byte-stream with Encryption Ratio
Lightweight Selective Encryption for Social Data Protection Based on EBCOT Coding
Outsourcing chaotic selective image encryption to the cloud with steganography
Edge-based lightweight image encryption using chaos-based reversible hidden transform and multiple-order discrete fractional cosine transform
Chaos-based efficient selective image encryption
Lightweight Selective Encryption for Social Data Protection Based on EBCOT Coding
Edge-based lightweight selective encryption scheme for digital medical images

Figure 2.1: Spatial Domain Image Encryption Categories

In-Compression approach

This approach requires a modification of the encoder and decoder system during an image byte stream (during compression) [11]. It results in an encrypted image. The compression process reduces and eliminates the duplicated blocks, making the encryption more secure. In other words, constant and similar blocks are reduced and then encrypted.

Coefficient correlation

Calculating the correlation of an image block helps define whether this block of an image contains important data. This important data is critical for concealing the whole image if all these blocks are encrypted. Depending on a threshold value, the calculated Coefficient Correlation of a block is either encrypted or left as it is. Encrypting these blocks conceals the important data of an image, making it visually indistinguishable.

Edge detection

Edge detection reveals the important visual information corresponding to discontinuities in the physical properties of an image [27]. In other words, sharp changes in colour or intensity of an image block help identify the important data of an image that makes an image distinguishable. Encrypting these areas tends to hide such information to make this important data concealed visually. This technique reduced the computational cost and time instead of encrypting all the image blocks.

Most Significant bits of an Image

The most significant bits (MSB) of the image pixels can be selected as the important data of an image. Since the MSB holds the most important data of a pixel, for example, the 8^{th} most significant bit of a pixel holds half the information ${2^7}_{256} \times 100\%$ and the 7^{th} bit holds 25% of the information ${2^6}_{100} \times 100\%$ and so on. Encrypting this part will lose the pixel colour (visually).

Byte-stream with Encryption Ratio

Another approach is to encrypt an image by ratio selectively. A byte stream of an image can be encrypted [22] within a specified ratio. Authors need to experimentally test their algorithm to reach the ultimate ratio percentage for encrypting the bit-steam to guarantee the robustness of image encryption.

2.3 Related Work

In this section, some of the recent and interesting research algorithms that address selective image encryption are described. As an example of chaotic encryption algorithms [14] [26] [23] [24] [16] are discussed, whereas in [21] Pseudo Random Number Generator (PRNG) is used during its presented encryption process.

2.3.1 Chaos Based Efficient Selective Image Encryption

In [14], the authors presented a lightweight, secure encryption scheme for digital images. The presented scheme starts by dividing plaintext images into several blocks, and correlation coefficient values are calculated for each block. Then, the blocks with the maximum values of correlation coefficients (C.C) are encrypted by XORing pixel-wise with random numbers generated from a skew tent map (based on a predefined threshold value). Finally, using two random sequences generated from the TD-ERCS chaotic map, the whole image is permuted. For confusion, the final encrypted image is shuffled row-wise and column-wise, respectively. Last decade, the use of social networks has significantly increased the demand for sharing multimedia data. Consequently, many algorithms have been developed to increase its security and difficulty against eavesdropping attacks. However, this has increased the computational cost and communication overhead and does not yet provide security against new zero-day attacks. This has motivated researchers to propose algorithms against these issues for better security and performance (cost).

The presented scheme can be summarized with the following steps (illustrated in Figure 2.2 [14]):

1. First, divide the plain text image into blocks $B = [B_1, B_2, \ldots, B_{256}]$ with total blocks of 256.
2. Then, define a threshold value and calculate the correlation coefficient of each block.
3. Each block of the plain-text image having a correlation coefficient (C.C) greater than the predefined threshold value (T=0.3) is bit-wised XOR-

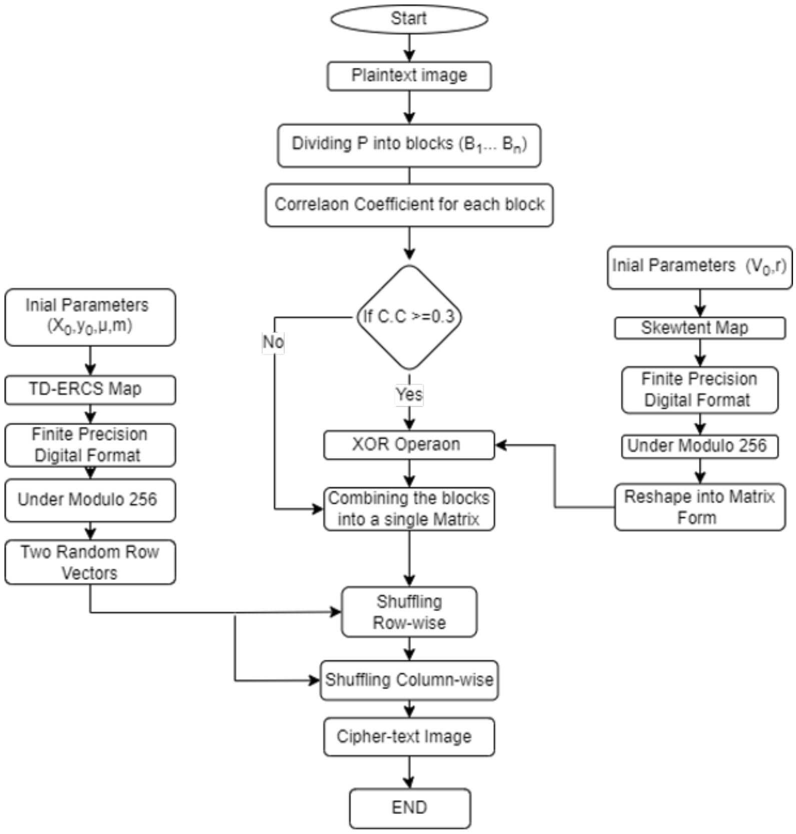


Figure 2.2: Flow Chart Of The Design Scheme

ed with a random number matrix ψ. Matrix ψ is obtained by arranging vector ζ in matrix ψ, where $\zeta = Module(Y, 256)$ by which is defined by $Y = V \times 10^{14}$ and V is a random vector generated by iterating V_{n+1} 65,536 times such that $t = 0.1000$ and $V_0 = 0.5000$ are the initial conditions for Skew Tent Map:

$$V_{n+1} = f(V_n, r) \tag{2.1}$$

$$V_{n+1} = \begin{cases} \frac{V_n}{r}, & if V_n \in [0, r] \\ \frac{1-V_n}{1-r}, & if V_n \in (r, 1] \end{cases} \tag{2.2}$$

4. Diffused blocks $Diff_{Block_n}$ are now generated $Diff_{Block_n} = bitxor(B_n, \psi)$ where n is the block (B_n) number undergoing the XOR operation.

5. All blocks are combined to get the diffused image $Diff_{image}$.

6. The row-wise permutation and then column-wise permutation are done on the diffused image using two random numbers generated $X^{"}$ and $Y^{"}$. These two numbers are generated using the mathematical representation of the Tangent Delay Ellipse Reflecting Cavity Map System (TD-ERCS)

$$X = X^1, X^2, ..., X^{256} \tag{2.3}$$

$$Y = Y^1, Y^2, ..., Y^{256} \tag{2.4}$$

7. Finally, the ciphertext image is obtained.

8. To decrypt the cypher image, all previous steps are applied reversely.

2.3.2 Edge-based Lightweight Image Encryption Using Chaos-based Reversible Hidden Transform and Multiple-order Discrete Fractional Cosine Transform

In [26], the authors proposed an encryption scheme consisting of edge detection (based on advanced cellular neural network structure “CNN”), chaos-based reversible hidden transform, and multiple-order discrete fractional cosine transform. This scheme encrypts the image regions with semantic information, whereas the other smooth regions will not be encrypted. About fifty per cent of image blocks were fully encrypted; thus, the computation cost was decreased.

Most image encryption algorithms encrypt all the image blocks regardless of contour features or other semantic information in an image; they only consider pixels and bits. This requires a high computational cost. Motivated by this, the authors propose in this paper a lightweight scheme that has a low computational cost that encrypts only contour features and other semantic information of an image. Moreover, transmitting a fully encrypted image after compressing it can result in a loss of compression ratio (to some extent). This loss can be remitted to some extent with the authors’ presented encryption scheme.

Encryption can be categorized into two categories: full and selective (partial) encryption. Full encryption encrypts the whole information, whereas selective encrypts a particular bit stream. The major difference between selective and full is the computational cost, as it is higher in full encryption. Even though selective encryption has a trade-off between security and complexity, it has wider practical applications.

In [26], the authors presented a new scheme for lightweight image encryption that can be summarized with the following: The image in the preprocessing step undergoes an edge detection step; it is an essential step to recognize significant contour features. Authors used edge detection based on Cellular Neural Network (CNN) with low computational cost. Then, the identified significant blocks are encrypted using Cross Chaotic Map-based Reversible Hidden Transform (CCM-based RHT) and Multiple-Order Discrete Cosine Transform (MODFrCT).

RHT transforms (maps) a pair to another one at a lower computational complexity such that:

$$y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \tag{2.5}$$

$$y = \begin{bmatrix} \alpha x_1 + \beta x_2 \\ \beta x_1 + \alpha x_2 \end{bmatrix} \tag{2.6}$$

where $y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$ is the transformed pair of pixels. And its inverse transform:

$$\tilde{X} \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} \tag{2.7}$$

$$\tilde{X} = \begin{bmatrix} \frac{\alpha Y_1 + \beta Y_2}{\alpha^2 - \beta^2} \\ \frac{\beta Y_1 + \alpha Y_2}{\beta^2 - \alpha^2} \end{bmatrix} \tag{2.8}$$

such that α and β are secret parameters and it changes for each image where $\alpha + \beta = 1$ and $0 \leq \alpha$, $\beta \geq 1$.

The authors illustrated the encryption/decryption process as shown in Figure 2.3 [26]:

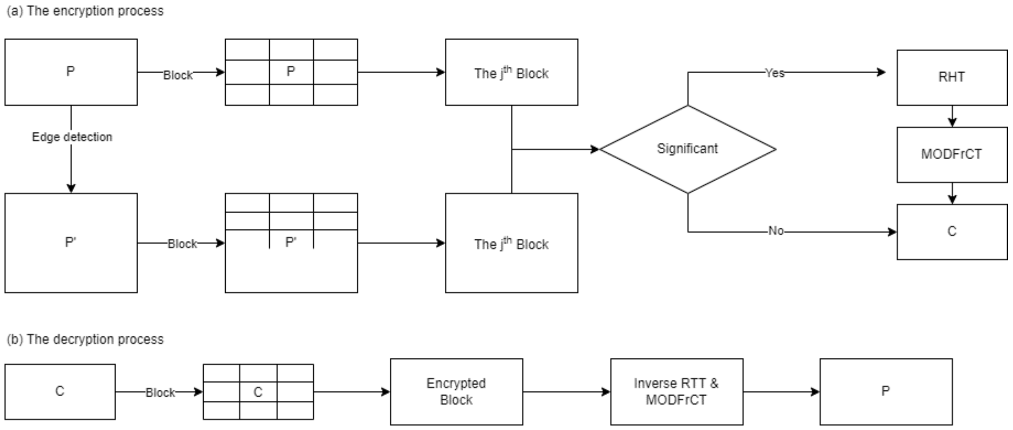


Figure 2.3: Block Diagram (a) The Encryption Process And (b) Decryption Process

To encrypt an image, they needed to:

1. Input a grey-scale image P with size $N \times N$:

   (a) Use an Edge detector to recognize the significance of each block.

   (b) From the original image P, generate the output of the detected edge as a binary image $P^{'}$. In binary image $P^{'}$, the detected pixel is referred to as "1" and "0" for the otherwise.

   (c) Divide P and P into non-overlapping $m \times m$ pixels blocks. The number of detected blocks in an image is $n = (N/m)^2$.

   (d) Calculate the significant degree of each detected pixel $\gamma_i = d_i/m^2$ of each block such that $d_i = 1, 2, ..., n$ and d_i is the detected pixels of a block i.

   (e) Set threshold level $(T, (0 < T \leq 1)$ and obtain Binary Significant Vector (BSV) for each block. The BSV value "1" indicates a significant block while "0" for the contrary. This BSV is required to decrypt the image on the receiver side.

2. Generate the keystream used in the Reversible Hidden Transform (RHT) and Multiple-Order Discrete Fractional Cosine Transform (MODFrCT) based on the Cross Chaotic Map (CCM).

   (a) Get the number of the significant blocks ϕ, and iterate CCM ϕ times with initial conditions a_0 and b_0 to obtain two key vectors a and b of length ϕ,

   (b) Generate keystream α_j and β_j such that $j = 1, 2, ..., \phi$ with the equation $a_j = (|a_j| + |b_j|)/2$.

   (c) Set $(|a_j| + |b_j|) \in (0, 2)$ as the orders of the MODFrCT, namely $P_j = |a_j| + |b_j|$.

3. Encrypt the significant blocks in sequence:

   (a) The j^{th} significant block is then transformed by the RHT with the corresponding parameters a_j and b_j.

   (b) Perform MODFrCT of the j^{th} sequence with the corresponding order P_j. Then, each sequence is replaced with the original block in the same position.

4. The final encrypted image is produced.

The decryption process is much simpler, as the receiver needs to have BSV to perform the inverse of MODFrCT and RHT.

The authors experimented using typical images such as Lena, Aerial, Boat, Goldhill, Baboon, Peppers, and woman. They found significant contour features in these images have been largely hidden.

2.3.3 Ievca An Efficient Image Encryption Technique For Iot Applications Using 2-D Von-Neumann Cellular Automata

Authors in [23] presented a lightweight encryption algorithm using 2-D Von-Neumann Cellular Automata (2VCA) with five neighbours, called IEVCA. It has all the properties of a good image cypher, including correlation immunity and lossless. It has passed all the randomness tests of DIEHARD and NIST statistical test suites.

It achieves a high level of diffusion and confusion by using pixel substitution of colour images. IEVCA is robust and highly secure as it has successfully undergone the security and performance analysis that conventional ones apply for.

Limited resource devices such as the Internet of Things (IoT) work as sensors sending images through the internet to cloud storage for further processing. Critical applications such as defence, and healthcare, require images to be encrypted before transmitting them to the public network to gateway fog nodes. Since conventional encryption algorithms cannot be deployed due to resource-limited devices. Motivated by this, authors in [23] presented encryption based on Cellular Automata (CA) for image encryption due to its simplicity in implementation, efficiency, and resistance to security attacks.

The presented scheme is implemented in the physical layer of a three-layer IoT deployment scenario; see Figure 2.4. The three layers are the physical layer where sensors reside and send the capture data to the above layer (network layer), the network layer where fog nodes exist, and they restrict massive network traffic towards the upper application layer (eliminating unnecessary traffic), and application layer where high-end computers and server cloud

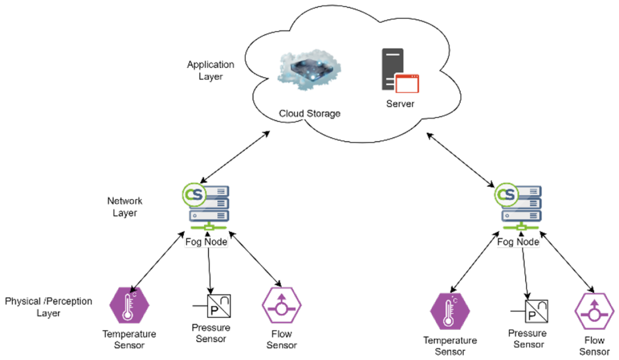


Figure 2.4: Three-Layer IoT Architecture

storage receive the sensors' data or analytical results. End users interact with the application layer.

Cellular Automata is a mathematical model consisting of simple components that act together under transformation rules to construct a complex system. CA can be achieved within many dimensions, such as 1-D CA and 2-D CA. In the presented encryption algorithm, 2-D von Neumann CA (2VCA) was considered. Periodic-boundary and null boundary considerations are used in the presented 2VCA IEVCA work. The extreme boundary cells in the periodic boundary are considered adjacent to each other while finding neighbours. Meanwhile, the extreme boundary cells in the null boundary are connected to the logic "Zero".

2-D CA rule generation

A cell in CA is transformed from 0 to 1 or vice versa according to specific CA rules. These rules are affected by the nine neighbours of a cell (including the cell itself). Most of the CA rules are constructed through primary rules. As an example of a primary rule, if a cell value is 0 and has at least three alive neighbours, then that cell's value is changed to 1. This rule is a primary rule, and other rules are generated using these primary rules. For example, let Rule 1, Rule 2, Rule 4, Rule 8, and Rule 16 be primitive rules such that:

$$Rule1 :: [S_t] = [S_t] \tag{2.9}$$

$$Rule1 :: [S_{(t+1)}] = [S_t][M_2] \tag{2.10}$$

$$Rule4 :: [S_{(t+1)}] = [M_2] \tag{2.11}$$

$$Rule8 :: [S_{(t+1)}] = [S_t][M_1] \tag{2.12}$$

$$Rule8 :: [S_{(t+1)}] = [S_t][M_1] \tag{2.13}$$

And M_1 , M_2 be any matrices for instance: $M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$, $M_2 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and $S_{(t+1}$ be the start of a cell at time $t + 1$ and S_t be the start of a cell at time t. More complex rules can be generated from the above primary rules, such as:

$$\begin{aligned} &Rule11P = Rule1P + Rule2P + Rule8P \\ &\therefore [S_{t+1}] = [S_t] + [S_t][M_2] + [S_t][M_1] \end{aligned} \tag{2.14}$$

$$\begin{aligned} &Rule24P = Rule8P + Rule16P \\ &\therefore [S_{t+1}] = [S_t][M_1] + [M_2][S_t] \end{aligned} \tag{2.15}$$

$$\begin{aligned} &Rule25P = Rule1P + Rule8P + Rule16P \\ &\therefore [S_{t+1}] = [S_t] + [S_t][M_1] + [M_2][S_t] \end{aligned} \tag{2.16}$$

$$\begin{aligned} &RuleRule30P = Rule2P + Rule4P + Rule8P + Rule16P \\ &\therefore [S_{t+1}] = [S_t][M_2] + [M_1][S_t] + [S_t] + [S_t][M_1] \end{aligned} \tag{2.17}$$

Note: P indicates the cells are under periodic boundary conditions, whereas N is for null-boundary conditions.

A rule vector (CARV) contains a set of rules. As an example, it can be symbolized such as:

$$CARV = \begin{pmatrix} 31N & 11N & 22N \\ 62N & 2N & 26N \\ 7N & 15N & 26N \end{pmatrix} \tag{2.18}$$

In the case of a group of cells having k as the initial configuration and when it undergoes a certain number of transitions and ends up with the initial state k, this CA is called Group CA (GCA). In this authors' presented work, Group CA (GCA) is obtained from different rule vectors. Rule vectors belong to different classes to ensure a high degree of confusion and diffusion property in the cypher images. Authors generated Von Neumann GCA rule vectors under both null-boundary and periodic-boundary conditions.

Image Encryption Algorithm

The encryption algorithm generates the encrypted image I_{enc} and the secret "symmetric" key K (to be used in the decryption algorithm). The encryption algorithm encrypts an input colour image (Figure 2.5) [23] I of size m×n using 2D CA rule vectors (CARVs). First, red R(MXN), green G(MXN) and blue B(MXN) channels are extracted from the plain colour image I. Then each channel $R(mxn), B(mxn), G(mxn)$ is converted into binary format; each pixel is converted into 8 bits "$Bin(R)(mx(8n), Bin(B)(mx(8n), Bin(G)(mx(8n)$"
.

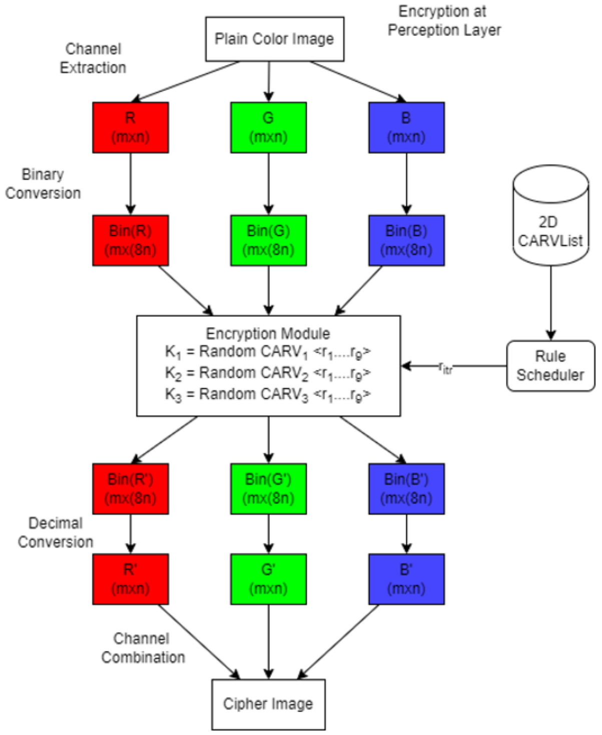


Figure 2.5: Encryption Process

Binary image blocks are substituted using the CARVs rule list. From this CARV list, three rule vectors (k_1, k_2, k_3) are selected randomly to encrypt these three channels (red, green, and blue). The encryption is done with a random round iterations (r_{itr}). In each iteration, the binary image goes through random CARVs taken from CARVList through the Rule scheduler.

Then, these binary images are converted into red, green, and blue channels to be combined as one encrypted image. The decryption process uses the same secret key, "K", for decrypting I_{enc}. It follows the same steps as in the encryption process, except it decrypts the encrypted binary images.

2.3.4 Lightweight Selective Encryption for Social Data Protection Based on EBCOT Coding

To effectively protect social media while its data is being generated, sent, transmitted, and shared through online social media platforms, the authors presented a novel design based on an agnostic selective encryption concept based on the embedded block coding with optimized truncation (EBOCT) system. Inspired by SE, the authors presented an effective agnostic selective encryption that encrypts a small subset of the byte-stream 8% of the stream) based on arithmetic techniques.

The trending development of social sensing systems has brought the urge to protect data while being generated, stored, and transmitted. Meanwhile, most traditional encryption algorithms are unsuitable for data protection in social sensing and data-sharing systems. Traditional encryption methods are designed based on the assumption of having one sender and one receiver during the communication process. However, this is not efficient when many users are involved as receivers. Besides that, existing Selective Encryption (SE) methods are unsuitable for today's social sensing data since they are strictly format reliance and implementing them on such data is very costly. Motivated by this, authors in [21] presented a selective encryption scheme based on an in-compression approach.

To protect social data generated, sent, and transmitted through social sensing systems, authors presented a novel design in [21]; they adopted an agnostic selective encryption concept based on the embedded block coding with an optimized truncation system (EBCOT).

During the EBCOT coding process, redundant data content is removed; thus, data will be very sensitive to the tiny changes since there is propagation for the decoding process such that a small ratio of SE could lead to very different output results, which can resist recovery from the attackers.

Authors presented this basic design to selectively encrypt in a lightweight manner some bitstream in the middle of the coding system process such that the output data files are protected.

The authors used the arithmetic coding system in JPEG2000 to propose SE encryption. The MQ coder in this arithmetic coding system is a context-based adaptive binary arithmetic coding system (AC). Two tiers mainly form the JPEG200 standard; tier-1 is the entropy coding combined with the MQ coding, and tier-2 is the packetization process, which generates the code packets from code streams. The authors adopted a tier-1 coding system from JPEG2000 as a compression process of the bitstream, and within this tier, they performed the Selective Encryption (SE) process with a percentage ratio of the bitstream.

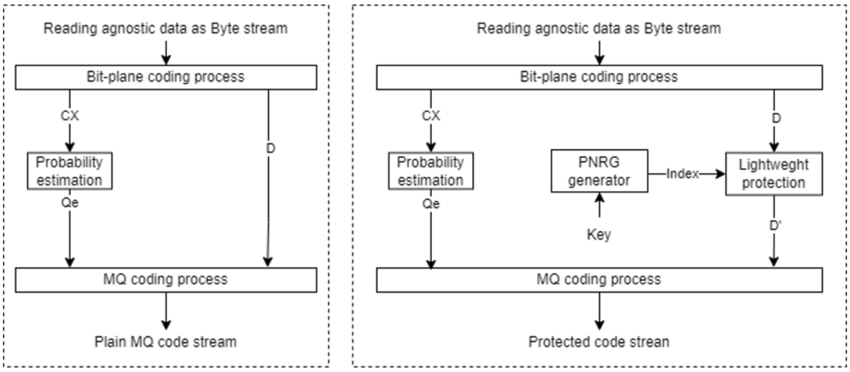


Figure 2.6: Architecture Of How The Coding-based SE Is Performed. (a) On Left Is The Normal MQ Coding Process. (b) On Right, Presented SE With The MQ Coding Process

As illustrated in figure 2.6 [21], data are read as byte streams agnostically, disregarding their original format. Then, the byte stream is encoded into the context information (CX) using a bit-plane encoding process. The MQ coding process uses the context information (CX) to control the adaptivity of AC by generating the probability estimation (Q_e) from the CX. In bitstream D, authors adopted an SM2LSB-plane encoder to selectively encrypt and protect "D" using a lightweight selective encryption algorithm [21] symbolized by SE_C function:

$$SE_C(F, K, R) \tag{2.19}$$

Where F is the data content representing the data input, K is a secret key, and R is the selection ratio.

Authors fragmented data context "F" bitstream into N fragmentations such that $F = F_1, F_2, F_3, ..., F_3$ for parallel processing encryption as displayed in Figure 2.7 [21]. Each fragment is selectively encrypted within the compression system in tier-1 by an algorithm summarized with $Enc_{(F)} = SE_C(F, K, R)$, where "K": is the secret key used for pseudorandom number generator (PRNG), "R:" is the ratio of selective encryption of bit steam, and "F:" is a fragment of data content.

The authors obtained the proper ratio "R" through the protection analysis test, in which most tests concluded a ratio of 8% with a high level of protection. The authors used the secret key "K" in" the pseudorandom number generator to selectively protect the bit stream. However, if this secret key is reused, the generated random number will be repeated (same), and it will be exposed to a chosen/known plaintext attack. Thus, the authors increased the inputs of PRNG to include three other parameters: the secret key "K",

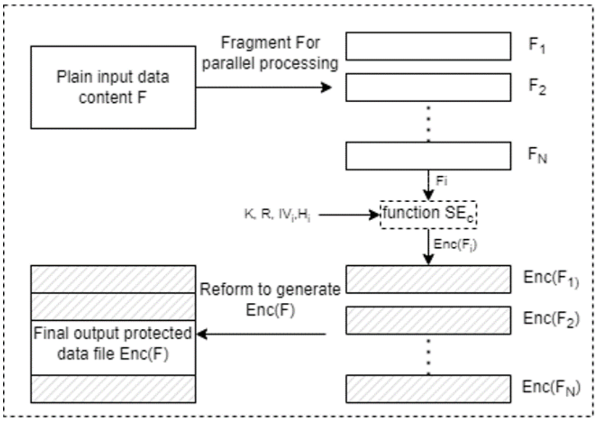

Plain input data content F
Fragment For parallel processing
F_1
F_2
F_N
Fi
K, R, IV_i,H_i
function SE_c
Enc(F_i)
Enc(F_1)
Enc(F_2)
Enc(F_N)
Reform to generate Enc(F)
Final output protected data file Enc(F)

Figure 2.7: System architecture of how the fragmentation of F is processed

the hash value of the input plaintext fragment "H" and the initialization vector "IV".

2.3.5 Outsourcing Chaotic Selective Image Encryption to The Cloud with Steganography

With the help of steganography, authors in [24] proposed a scheme for outsourcing chaotic selective image encryption to the cloud to protect image data from being exposed to a third-party cloud service.

Devices such as smartphones and real-time communication devices face a challenge in encrypting images because of their limited resources (energy and computational power) thus traditional encryption paradigm (entire bit stream image encryption) is no longer suitable. Besides that, image encryption outsourcing also encounters other challenges on how to protect data images and not be revealed to third-party cloud outsourced encryption services. Many researchers have been seeking chaotic encryption for image encryption, even though many chaotic image cyphers are computationally extensive (cannot be managed with resource-limited devices). Motivated by this, the authors considered the problem of selectively encrypting a plain image by a chaotic map and distributing the encrypted image to other users with no sufficient computational power or energy supply to be outsourced on the cloud.

In the authors' scheme, a resource-limited client sends a stego image (contains selective secret information image) to a cloud service to do the chaotic encryption. The Cloud service sends the encrypted stego image back to the user. The user shares encrypted images with other users. In [24], the presented scheme is shown in Figure 2.8 [24].

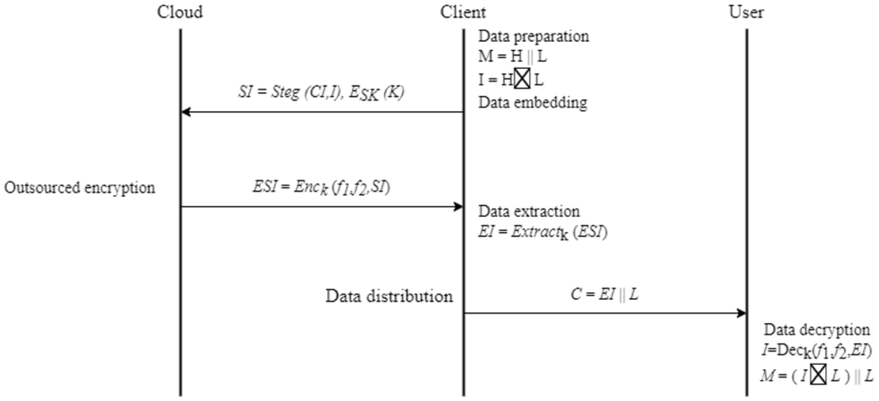


Figure 2.8: The Framework Of The Presented Scheme

A client chooses the important data part in an image M of size "m×n" to be selectively encrypted. The image "M" can be constructed as the 4 most significant bits (MSBs) H and the 4 least significant bits L (LSBs), i.e., $M = H||L$. Important information (I) of the plain image is masked by doing the XORs with L such that $I = H \oplus L$.

Then I is embedded into a cover image (CI) using Single Match 2 Least Significant Bit (SM2LSB) to produce a stego image (SI) so that (SI=Steg(CI, I). Then, the client sends a stego image (SI) to the cloud for chaotic encryption along with the encrypted key (K) using a fast encryption cypher with the shared key (SK). Cloud does not know of the existence of hidden data.

In the cloud, the stego image is encrypted using two predetermined chaotic maps f_1, f_2, i.e. $ESI = Enc_k(f_1, f_2, SI)$; f_1 to permutate the pixel position of SI and f_2 to perform XOR masking with each pixel value of the permutated (SI). The cloud encrypts the whole stego image SI and sends it back to the client. The client extracts the encrypted embedded important data (EI), such that $EI = Extract_K(ESI)$

Now, the client can transfer the selectively encrypted image (C) to other users securely through public channels. The client needs to get the encrypted image (C) by concatenating (EI) and the least significant bit L such that $C = EI||L$.

The selectively encrypted image (C) can be decrypted by obtaining the secret key (K) and running the decryption algorithm to get the important information "I" such, i.e. $I = Dec_K(f_1, f_2, EI)$. The client can know which data (EI) is selectively encrypted by splitting C into EI and L such that $C = EI||L$. After obtaining the important information "I", the user can easily get the decrypted image M by $M = (I \oplus L)||L$.

Authors adopted SM2LSB presented in [13] since it offers lower probability detection of hidden important data by making fewer changes to the cover image compared to 2LSB replacement. The main idea behind Single Match 2LSB is to embed 2-bit information into 2 LSB of the cover image, and a third LSB is used as a flag indicating the mismatch position.

SM2LSB maximizes embedding capability while keeping high security in the spatial domain by reorganizing and embedding important data to a cover image. In data reorganization, important data (I) containing 4 bitplanes is reorganized into 2 bitplanes of I', and by doing so, the vulnerability of having consecutive 0s and 1s in I' is avoided. The size of I' is the same as the used cover image (to support chaotic encryption in a later step). It reorganizes 4 bitplanes of I into 2 bitplanes by extracting each bit of 4 bitplanes in a repeated raster scan order for all the coordination of I and buffering the extracted bits onto a sequence. The authors used this sequence to construct the first and second bitplanes of I'. In data embedding, SM2LSB ensures the minimal changes of 2LSB in the cover image for much lower detection than 2LSB.

The receiver decrypts using key $Dec_{SK}(K)$ the 2LSB of ESI and partially decrypts the third LSB (flag to indicate the position of mismatch bit). This finally results in obtaining the hidden decrypted data.

2.3.6 Edge-Based Lightweight Selective Encryption Scheme for Digital Medical Images

To overcome the computation complexity and high processing time, the authors in [16] adopted an edge-based lightweight selective encryption in their work. They used a combination of One Time Pad (OTP), edge detection (Prewitt edge) and a Chaotic map approach. Authors used edge detection only to encrypt the significant image blocks, thus reducing the computational time using the OTP algorithm. To resist well-known attacks, the authors used the chaotic map in [16] to produce a highly sensitive key with an appropriate large key space and, at the same time, have a relatively high image quality.

The rapid growth and the storing of transferred medical images through the public network have a high demand to secure such traffic, considering the special structure of these medical images. Traditional algorithms are designed for textual data and not for images, which have their complexity. Medical images in large data volumes correlate strongly between pixels and have high redundancy.

Chaos-based encryption has also been considered in the authors' work in [16]. Since chaotic encryption can efficiently and securely encrypt images due to the randomness of its output. However, chaotic encryption still has computation complexity and high processing time, especially in real-time applications. Motivated by this, the authors have proposed selective encryption of the medical images in [16] to overcome the mentioned issues.

In [16], the authors presented a scheme (illustrated in Figure 2.9 [16]) that starts by decomposing a medical image into non-overlapping blocks of pixels of a specific size. Then, using Prewitt edge detection, significant image blocks were identified according to a specific threshold value. Then, using the chaotic map, a matrix of random keys is generated that equals the total number of significant blocks in an image. Finally, each identified significant block is encrypted in sequence using the one-time pad algorithm.

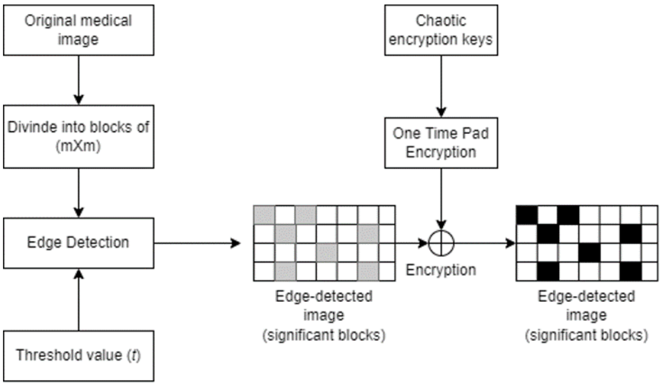


Figure 2.9: Block Diagram For Edge-Based Medical Image Encryption

Chapter 3

Proposed Model

3.1 Introduction

Our proposed algorithm targets sensitive bits and pixels of the region of interest through pixel scrambling, exclusive-OR operation and red, green and blue (RGB) channels swapping. The most significant bit of a pixel is considered a sensitive bit since this 8^{th} bit holds 50% of the pixel's value. Moreover, the other three significant bits (7^{th}, 6^{th}, and 5^{th}) can also be considered as sensitive bits.

The proposed algorithm divides the image into n number of blocks and only encrypts the region of interest. First, the Laplacian edge detector determines which blocks contain the critical data. Then, the encryption algorithm takes place on these crucial blocks. Thus, the proposed algorithm divides blocks' encryption into three phases. The first phase encrypts the most significant bit of each byte within the three channels using an exclusive OR operation. Second, it swaps the other three most significant bits among the RGB channels. Finally, it scrambles the pixels row-wise and executes an exclusive-OR operation over all the blocks' pixels. We utilize the cypher block chaining

(CBC) mode of operation to prevent the region of interest from being statistically distinguishable.

3.2 Data-set

This section presents the data sets utilised for our experimentation and the evaluation metrics employed. We used Lena and Barbara's images [1] [2]; both images are colour and have a size of 512 × 512 pixels. Moreover, we utilized an online Pseudo Random Number Generator (PRNG) [3] within our encryption/decryption algorithms.

3.3 Proposed Algorithm

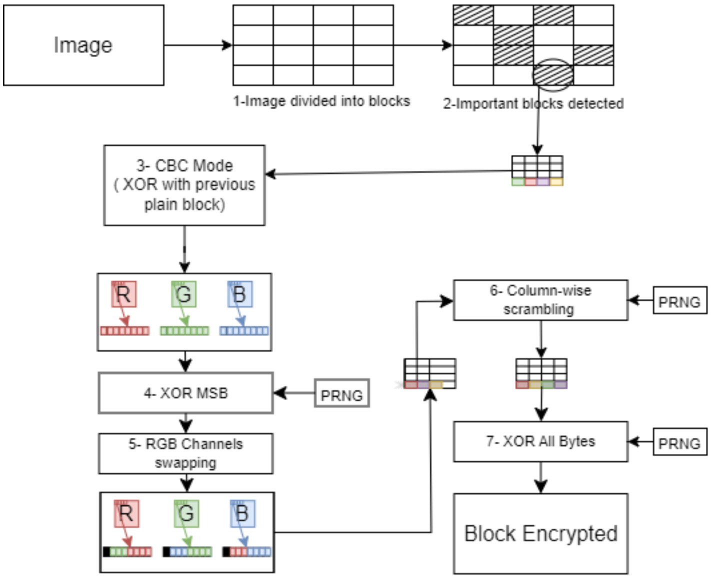


Image
1-Image divided into blocks
2-Important blocks detected
3- CBC Mode (XOR with previous plain block)
R
G
B
4- XOR MSB
PRNG
5- RGB Channels swapping
R
G
B
6- Column-wise scrambling
PRNG
7- XOR All Bytes
PRNG
Block Encrypted

Figure 3.1: Proposed Encryption Algorithm

As illustrated in Figure 3.1, our proposed solution involves selectively encrypting an image by encrypting only the blocks' region of interest. We determine the ROI blocks using the Laplacian edge detection method. The image is initially divided into multiple blocks, and we determine the crucial blocks based on their edge count (blocks having an edge count exceeding a specific threshold). Subsequently, using an exclusive-OR operation, we encrypt the most significant bit (MSB) of each pixel's three channels within the selected blocks. Once we have encrypted the MSB bit of the three channels within the blocks' pixels, we swap and scramble. The swapping is done among the pixels' RGB channels, and the scrambling is done among the pixels within a block column-wise (using the Fisher-Yates algorithm). Finally, All pixels (bytes) of our region of interest are encrypted using exclusive-OR operation. Moreover, to guarantee the robustness of the proposed encryption scheme, cypher block chaining mode is implemented on the clear text blocks before being encrypted.

The proposed encryption algorithm, as illustrated in Figure 3.1 and pseudo-coded in Algorithm 1, can be summarised by the following points:

1. Divide the image into blocks
2. Identify the ROI using Laplacian edge detection
3. Execute exclusive-OR operation over the plain block with the previous cypher block/initial value (CBC mode of operation).
4. Execute exclusive-OR operation on the most significant bit
5. Swap the three other most significant bits among the RGB channels
6. Scramble pixels within the block column-wise using the Fisher-Yates algorithm
7. Execute exclusive-OR on all bytes

In the upcoming subsections, we will proceed with a detailed review of each of the aforementioned points in isolation.

Algorithm 1 Proposed Encryption Algorithm

```
Require: $image$
  $Blocks \leftarrow image\_divider(image)$
  $enc\_blocks \leftarrow image\_divider(image)$
  $counter \leftarrow 0$ for each $block \in Blocks$ do
          $counter++$
          $edge\_count = get\_laplacian\_edge\_count(block)$
          If $edge\_count \geq threshold$ then
              $enc\_block[counter] = cbc\_mode(block)$
              $enc\_block[counter] = msb\_xor(enc\_block[counter])$
              $enc\_block[counter] = scramble\_pixels(enc\_block[counter])$
              $enc\_block[counter] = xor\_bytes(enc\_block[counter])$
          end if
  end for $row < Block.rows$
```

Algorithm 2 Proposed Decryption Algorithm

```
Require: $enc\_image$
  $Blocks \leftarrow image\_divider(enc\_image)$
  $dec\_blocks \leftarrow image\_divider(enc\_image)$
  $counter \leftarrow 0$ for each $block \in Blocks$ do
          $counter++$
          $edge\_count = get\_laplacian\_edge\_count(block)$
          If $edge\_count \geq threshold$ then
              $dec\_block[counter] = xor\_bytes(enc\_block[counter])$
              $dec\_block[counter] = scramble\_pixels(enc\_block[counter])$
              $dec\_block[counter] = msb\_xor(enc\_block[counter])$
              $dec\_block[counter] = cbc\_mode(block)$
          end if
  end for $row < Block.rows$
```

(a) Lena Image

(b) Lena Image - Blocks

Figure 3.2: Lena Image Blocks

Algorithm 2 serves as a summary for the decryption algorithm, which acts as the reverse algorithm of the proposed encryption algorithm outlined in Algorithm 1.

3.3.1 Divide the image into blocks

A plaintext image has been divided into multiple blocks (n) to identify the significant blocks of the image (Regions Of Interest). For instance, Lena's 512x512 pixel image has been separated into 8x8 blocks, each containing 64x64 pixels.

Figure 3.2a shows Lena's test image and is divided into 64 blocks, as demonstrated in Figure 3.2b.

3.3.2 Identify the important blocks using Laplacian edge detection

The Laplacian edge detector filters the image and highlights all edges with high contrast properties, as shown in Figure 3.3a. Additionally, Figure 3.3b displays the filtered image divided into 8 x 8 blocks. The average number of all edges in an image's blocks is counted and calculated using the following equation:

(a) Lena Image (Laplacian)

(b) Lena Image - Blocks (Laplacian)

Figure 3.3: Lena Image

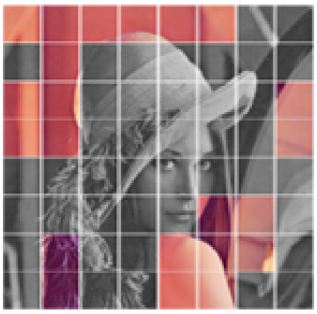


Figure 3.4: Lena Image (important blocks greyed out)

$$Avg_{EDGE} = \frac{\sum_{i=1}^{n} CountEDGE(block_i)}{n} \tag{3.1}$$

The Avg_{EDGE} is used as a criterion for selecting the most critical blocks (region of interest) in the image to be encrypted. A block having an edge count larger than Avg_{EDGE} is encrypted. The greyed-out blocks in Figure 3.4 represent the region of interest (ROI).

3.3.3 Cipher Block Chaining Mode Of Operation

Using cypher block chaining mode (CBC), each plain block of the ROI is XORed with the previously encrypted block. The first block is XORed with an initial vector (IV) - a randomly generated block. This IV is a single

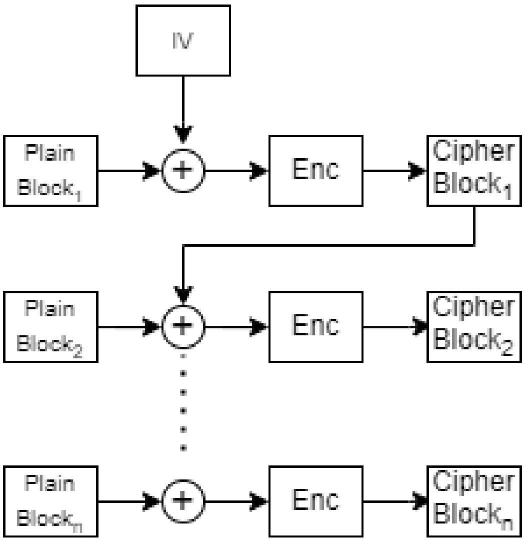


Figure 3.5: Block Images with CBC Mode Of Operation

uniformly distributed array of random numbers XORed with each pixel of the first plain block. Figure 3.5 illustrates the CBC mode used within our encryption algorithm.

3.3.4 Execute Exclusive-OR operation on the most significant bit

A coloured image consists of pixels constructed from three channels: red, green and blue (RGB). Each channel consists of 8 bits, as shown in Figure 3.6.

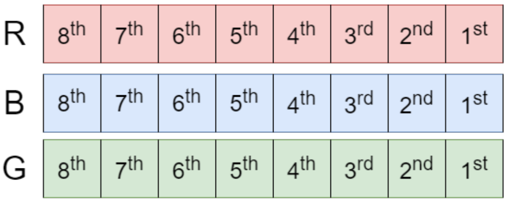
Once the region of interest is specified and CBC has taken place, the 8^{th} bit of each pixel of the RGB channels in the selected block is XORed using the following equations in 3.2:

$$\begin{aligned} MSB_{new_R} &= MSB_{old_R} \bigoplus IV_R \bigoplus Random\_Bit_R \\ MSB_{new_G} &= MSB_{old_G} \bigoplus IV_G \bigoplus Random\_Bit_G \\ MSB_{new_B} &= MSB_{old_B} \bigoplus IV_B \bigoplus Random\_Bit_B \end{aligned} \tag{3.2}$$

Where MSB_{new} is the resulting new bit value, $Random\_Bit$ is randomly generated bit by a pseudo-random number generator and IV_R, IV_G, and IV_B stand for the initial vectors of the RGB channels, respectively. These initial values have either one of these two values:

1. If there is no neighbouring pixel to the left of the encrypted pixel (the first pixel being XORed), it must have a predetermined value.

2. When there is a pixel adjacent to the encrypted pixel on the left, the IV values for each channel will be set to the previously XORed 8^{th} bit (the value of the previously XORed pixel MSB_{new}).



R	8th	7th	6th	5th	4th	3rd	2nd	1st
B	8th	7th	6th	5th	4th	3rd	2nd	1st
G	8th	7th	6th	5th	4th	3rd	2nd	1st

Figure 3.6: A Pixel (RGB Channels)

The MSB encryption of each RGB channel using exclusive-OR is demonstrated in Figure 3.7. The resulting image with the encrypted MSB channels is depicted in Figure 3.9 and is visually recognizable. To further enhance the encryption, the process involves swapping and shuffling (Fisher-Yates), as described in the following sections.

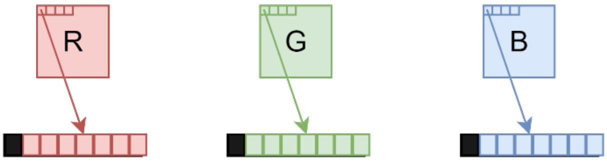


Figure 3.7: MSB encryption for each RGB channel

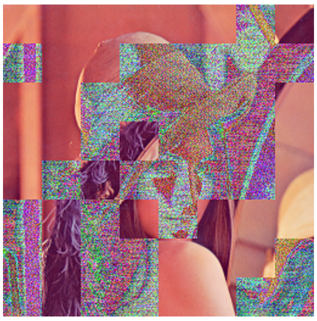


Figure 3.8: Lena with MSB encrypted - without CBC

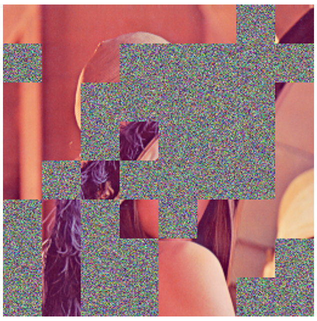


Figure 3.9: Lena with MSB encrypted - with CBC

3.3.5 Swap the other three most significant bits among the RGB channels

Even though the 8^{th} bit is encrypted, still, the image is still recognizable 3.8. The other three most significant bits are changed in a particular order; the swapping/mapping is done as the following equations (3.3, 3.4 and 3.5), and is illustrated in Figures 3.10, 3.11a and 3.11b.

Within the pixels of ROI, equation 3.3 swipes the 5^{th}, 6^{th} and 7^{th} bits of the Red channel (R) with the Green channel (G). Meanwhile, Equation 3.4 exchanges the 5^{th}, 6^{th} and 7^{th} bits of the Green channel (G) with the Blue channel (B), and so on for equation 3.5.

$$Map(R, G) = enc(8^{th}).G(6^{th}).G(5^{th}).G(7^{th}).R(5^{th}).R(4^{th}).R(3^{rd}).R(2^{nd}).R(1^{st}) \tag{3.3}$$

$$Map(G, B) = enc(8^{th}).B(6^{th}).B(5^{th}).B(7^{th}).G(5^{th}).G(4^{th}).G(3^{rd}).G(2^{nd}).G(1^{st}) \tag{3.4}$$

$$Map(B, R) = enc(8^{th}).R(6^{th}).R(5^{th}).R(7^{th}).B(5^{th}).B(4^{th}).B(3^{rd}).B(2^{nd}).B(1^{st}) \tag{3.5}$$

Figure 3.13 illustrates Lena's image with the 8^{th} MSB bit encrypted for each Pixel of the selected Block in the region of interest, and the other three

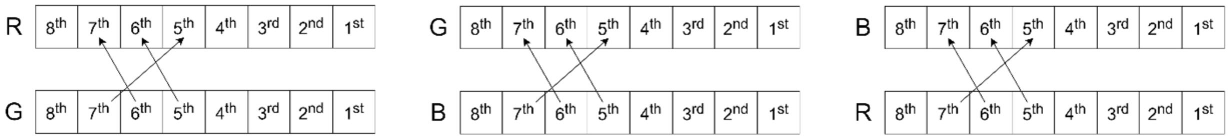
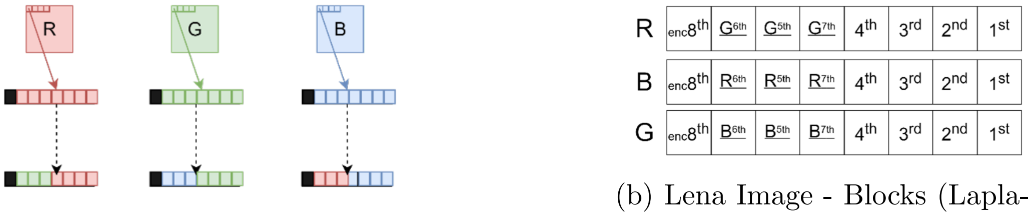

R 8th 7th 6th 5th 4th 3rd 2nd 1st
G 8th 7th 6th 5th 4th 3rd 2nd 1st
G 8th 7th 6th 5th 4th 3rd 2nd 1st
B 8th 7th 6th 5th 4th 3rd 2nd 1st
B 8th 7th 6th 5th 4th 3rd 2nd 1st
R 8th 7th 6th 5th 4th 3rd 2nd 1st

Figure 3.10: Bits swapping among the three channels (RGB)



(a) Lena Image (Laplacian)

R	enc8th	G6th	G5th	G7th	4th	3rd	2nd	1st
B	enc8th	R6th	R5th	R7th	4th	3rd	2nd	1st
G	enc8th	B6th	B5th	B7th	4th	3rd	2nd	1st

(b) Lena Image - Blocks (Laplacian)

Figure 3.11: Encrypted Lena Image

MSB bits are shuffled among the other channels.

3.3.6 Scramble pixels within the block column-wise using the Fisher-Yates algorithm

The next step is to shuffle the pixels (**column-wise**) within each Block holding the crucial data (ROI) using the Fisher-Yates algorithm. Figure 3.14 illustrates an example of the Fisher-Yates input and output.

Algorithm 3 illustrates the Fisher-Yates algorithm, which goes through each pixel within a row and shuffles them randomly, starting from the last pixel and ending at the first one (indexed 1). It swaps the value of a ran-

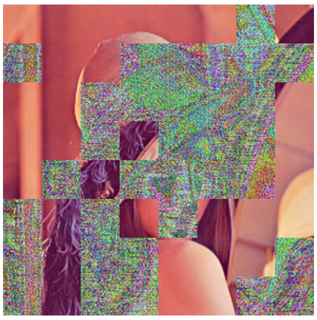


Figure 3.12: Lena image with MSB encryption and RGB pixels swapping - Without CBC

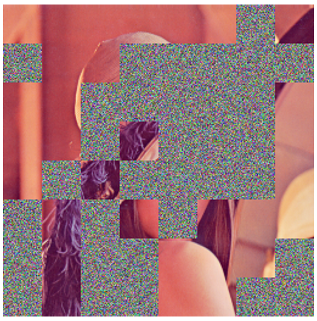


Figure 3.13: Lena image with MSB encryption and RGB pixels swapping - With CBC

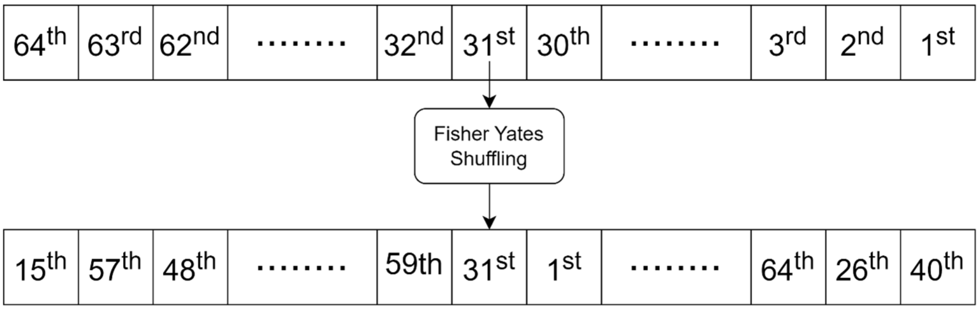

64th 63rd 62nd .... 32nd 31st 30th 3rd 2nd 1st
Fisher Yates Shuffling
15th 57th 48th .... 59th 31st 1st 64th 26th 40th

Figure 3.14: Fisher-Yates Algorithm Input-Output

domly selected pixel (its index is greater than the current one) within a row and replaces it with the current pixel in the for-loop. Figure 3.16 displays the scrambled pixels of Lena's image (column-wise).

3.3.7 Execute Exclusive OR on all bytes

In the last step of our algorithm, the algorithm applies exclusive-OR operation over all the regions of interest (ROI). Each channel within the pixels is XORed with an initial vector (IV) and a random byte producing new pixels of the being encrypted ROI as in equations 3.6.

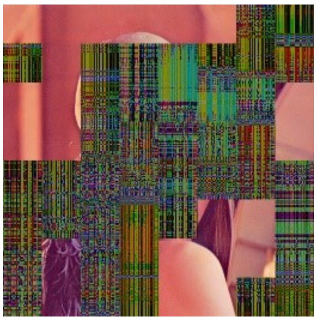


Figure 3.15: Lena's image after encryption and shuffling process - without CBC

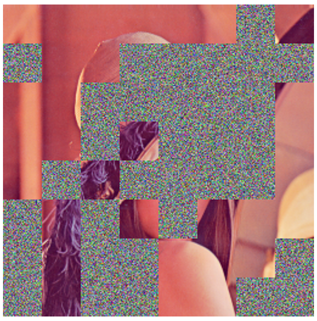


Figure 3.16: Lena's image after encryption and shuffling process - with CBC

$$\begin{aligned} Pixel_{new_R} &= Pixel_{old_R} \bigoplus IV_R \bigoplus Random\_Byte_R \\ Pixel_{new_G} &= Pixel_{old_G} \bigoplus IV_G \bigoplus Random\_Byte_G \\ Pixel_{new_B} &= Pixel_{old_B} \bigoplus IV_B \bigoplus Random\_Byte_B \end{aligned} \tag{3.6}$$

Where $Pixel_{new}$ is the result of the XOR operation over the current pixel, $Random\_Byte$ is randomly generated by a pseudo-random number generator and IV_R,IV_G, and IV_B stand for the initial vectors of the RGB channels. These initial values have either one of these two values:

1. If there is no neighbouring pixel to the left of the encrypted Pixel, it must have a predetermined value.

2. When there is a pixel adjacent to the encrypted Pixel on the left, the IV values for each channel will be set to the previously encrypted byte.

Algorithm 3 Fisher-Yates Algorithm

```
Require: $Block \in Image$
  $randon\_index \leftarrow 0$
  $current\_index \leftarrow 0$
  $previous\_randon\_index \leftarrow 0$
  $temp \leftarrow 0$
  for each $row \in Block.rows$ do
      for each $col \in Block.cols$ do
          If $col$ not equal to zero then
              $randon\_index \leftarrow prng()\%(cols + 1)$
          else
              $randon\_index \leftarrow 0$
          end if
          $previous\_randon\_index \leftarrow randon\_index$
          $current\_index \leftarrow col$
          $temp \leftarrow block.at(row, current\_index)$
          $block.at(row, current\_index) \leftarrow block.at(row, random\_index)$
          $block.at(row, random\_index) \leftarrow temp$
      Until $col \geq 0$
  Until $row < Block.rows$
```

Algorithm 4 represents the pseudo-code for this process, and Figure 3.18 demonstrates its output.

3.4 Conclusion

In conclusion, the outlined process presents a comprehensive method for image encryption. The image is initially divided into blocks, and the important blocks are identified using Laplacian edge detection. The Cipher Block Chaining Mode of Operation is then employed, followed by the execution of an exclusive-OR operation on the most significant bit. Subsequently, the other three most significant bits among the RGB channels are swapped. The Fisher-Yates algorithm is used to scramble pixels within the block column-wise. Finally, an exclusive OR operation is executed on all bytes. This method ensures a robust and secure encryption of images, enhancing data

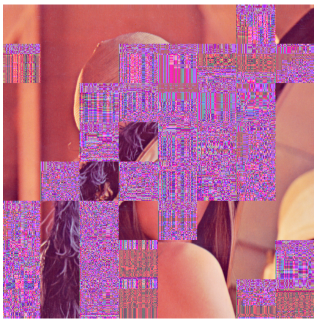


Figure 3.17: Lena image after byte exclusive-OR operation

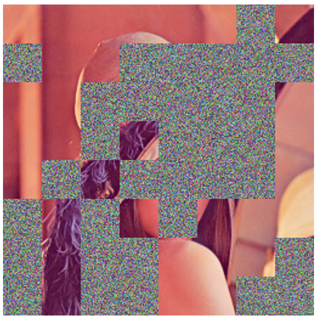


Figure 3.18: Lena image after byte exclusive-OR operation - with CBC

protection and privacy.

Algorithm 4 Byte's XOR Algorithm

```
Require: $plain\_block \in ROI$
  $randon\_byte\_r \leftarrow 0$
  $randon\_byte\_g \leftarrow 0$
  $randon\_byte\_b \leftarrow 0$
  $previous\_xored\_byte\_r \leftarrow 0$
  $previous\_xored\_byte\_g \leftarrow 0$
  $previous\_xored\_byte\_b \leftarrow 0$
  $iv\_byte\_r \leftarrow 0$
  $iv\_byte\_g \leftarrow 0$
  $iv\_byte\_b \leftarrow 0$
  for each $col \in plain\_block.cols$ do
      $previous\_encrypted\_byte\_r \leftarrow NULL$
      $previous\_encrypted\_byte\_g \leftarrow NULL$
      $previous\_encrypted\_byte\_b \leftarrow NULL$
      for each $col \in plain\_block.cols$ do
          $random\_byte\_r \leftarrow prng()\%255$
          $random\_byte\_g \leftarrow prng()\%255$
          $random\_byte\_b \leftarrow prng()\%255$
          If $col$ not equal to zero then
              $iv\_byte\_r \leftarrow previous\_encrypted\_byte\_r$
              $iv\_byte\_g \leftarrow previous\_encrypted\_byte\_g$
              $iv\_byte\_b \leftarrow previous\_encrypted\_byte\_b$
          else
              $iv\_byte\_r \leftarrow prng()\%255$
              $iv\_byte\_g \leftarrow prng()\%255$
              $iv\_byte\_b \leftarrow prng()\%255$
          end if
          xored_byte_r $\leftarrow$
                  $plain\_block.r\_channel\_at(row, col) \oplus iv\_byte\_r \oplus randon\_byte\_r$
          xored_byte_g $\leftarrow$
                  $plain\_block.g\_channel\_at(row, col) \oplus iv\_byte\_g \oplus randon\_byte\_g$
          xored_byte_b $\leftarrow$
                  $plain\_block.b\_channel\_at(row, col) \oplus iv\_byte\_b \oplus randon\_byte\_b$
           Until $col \geq 0$
  Until $row < Block.rows$
```

Chapter 4

Experiments and Results

4.1 Trials and errors

Throughout our thesis, we tried to get optimal results that could lead us to achieve a selective encryption methodology. Thus, different selection criteria were tested to distinguish the ROI of an image — MSB, low, high, square root, and edge criteria.

1. **MSB criterion**: All pixels in a block with reg, green and red values greater than 127 are counted as an MSB. So, all blocks with MSB count higher than an image's average MSB (equation 4.1) value are considered important blocks. Figure 4.1a illustrates the result of this criterion.

$$Avg_{MSB} = \frac{\sum_{i=1}^{n} CountMSB(block_i)}{n} \tag{4.1}$$

2. **Low criterion**: A block with MSB and EDGE count values less than an image's MSB and EDGE average (equations 4.1 and 3.1) is considered vital. Figure 4.1b demonstrates the result of this criterion.

3. **High criteria**: A block with MSB and EDGE count values more significant than an image's MSB and EDGE average (equations 4.1 and 3.1) is considered vital. (Figure 4.1c)

4. **Square root criterion** : it determines if the block has essential data according to the following equation:

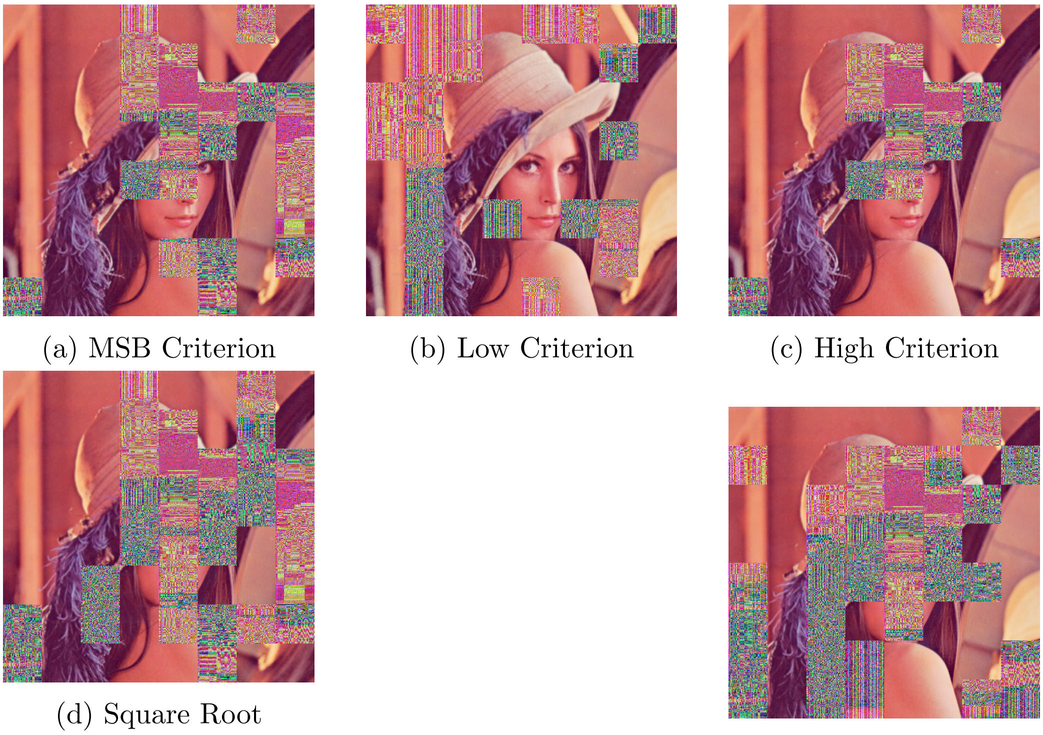
$$if(\sqrt{MSB(Block) + Avg_{EDGE}} > Avg_{MSB}) \tag{4.2}$$

$$encrypt(Block) \tag{4.3}$$

   The result of this criteria is shown in (Figure 4.1d).

5. **Edge criterion**: The Laplacian edge detector calculates all pixels' edge count. If the total count is more significant than an image's average EDGE (equation 3.1) value, it is considered part of ROI. The result of this criterion is shown in figure 4.1e.

Based on these test criteria, the edge criterion covers an image's required ROI. This criterion is implemented over Barbara's image, achieving the same results.



(a) MSB Criterion

(b) Low Criterion

(c) High Criterion

(d) Square Root Criterion

(e) Edge Criterion

Figure 4.1: Selection Criteria

4.2 Evaluation Metrics

Our algorithm is divided into distinct procedures, and each is developed and evaluated individually to acquire a strong encryption system. We meticulously evaluated each process, from performing exclusive-OR over the MSB to conducting the exclusive-OR operation over all bytes within each image block. We used abbreviations indicating each process to simplify the upcoming evaluation results in the abbreviation list (M, MC, MCP and MCPB).

In our evaluation metrics of the proposed scheme, it can be noticed the result values in full encryption in the MCPB process are near the optimal value contrary to selective encryption, which can sometimes be close to the optimal values. This can be justified due to not encrypting all the image blocks but only their selected essential blocks (region of interest). Even though omitting the unencrypted blocks from our calculations achieves the optimal values, obtaining the same results as shown for the fully encrypted images.

In this section, we have analyzed

1. Algorithm Complexity
2. Mean Square Error Function (RMSE) and Peak signal-to-noise ratio (PSNR)
3. Multi-Structural Similarity (MSSIM)
4. Key Sensitivity Test
5. Correlation Coefficient analysis (CC)
6. Information entropy analysis (IE)
7. Histogram of cypher image

8. Plaintext Sensitivity Test

9. Time Performance

4.2.1 Algorithm Complexity

Referring to Algorithm 1, our selection criteria for the two images have achieved about 50% reduction in encryption complexity (time and computation). The proposed algorithm encrypted 33 blocks in Lena's image out of 64 and 29 blocks in Barbara's Image - in other words, only 51.6% and 45.3% of the blocks were encrypted, respectively, for Lena and Barbara's images. Since Algorithm 1 visited only 50% of the blocks n, then its algorithm complexity is $\Omega(\frac{1}{2}n)$, thus achieving $\Omega(c.n)$ where c is a constant number less than 1.

Let's keep in mind, that for determining the edge count of the (8×8) blocks of our ROI, the algorithm must pass over all the blocks to calculate each edge count. This brings the algorithm complexity to $O(n)$.

4.2.2 Mean Square Error Function (MSE) and Peak signal-to-noise ratio (PSNR)

Mean Square Error Function (MSE)

The MSE (Mean Squared Error) measures the degree of similarity and distortion between an image and its encrypted image. It helps determine the reliability of encryption cryptosystem.[15]. A high value indicates an unrecognizable plain image from a cypher image; therefore, a cypher image cannot be deciphered into a plain image [23] [12].

The plain and decrypted image's pixels are $P(i,j)$ and $C(i,j)$, respectively. Where i,j denotes the pixel's location (at the i^{th} row and j^{th} column) of cypher and original images of size M × N, respectively.

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(i,j) - P(i,j)|^2 \tag{4.4}$$

Peak signal-to-noise ratio (PSNR)

PSNR indicates the quality of a cryptosystem scheme. It measures the difference between the original image and the encrypted one; The low PSNR value (near the value of 8.0) indicates a significant difference between the encrypted and original images [21]. The following equation 4.5 applies [4]:

$$PSNR = 10 \times log_{10}(\frac{255^2}{MSE}) \tag{4.5}$$

In our proposed work, it is observed from the results in tables 4.3 and 4.4 both Lena and Barbara's images achieved near-optimal MSE and PSNR values through selective and full encryption using CBC mode, meanwhile without CBC mode (tables 4.1 and 4.2) these values were less than the prior values but still near the optimal values. And throughout developing the algorithm, from performing exclusive-or operation over the most significant bits of the three RGB channels to performing it over all the bytes of the RGB channels, the PSNR values were getting closer to the optimal value as an indication of a more robust cryptosystem being developed.

In table 4.4, a comparison of our MSE and PSNR results with the work of [23] for full encryption with CBC Mode has shown our algorithm achieved a lower PSNR, indicating a more degree of randomness in the cypher image.

Table 4.1: Mean Square Root (MSE) and Peak signal-to-noise ratio (PSNR) - **Selective Encryption without CBC**

		MSE			PSNR		
Process	Image	Red	Green	Blue	Red	Green	Blue
M^*	Lena	4230.5	4265.68	**4034.87**	11.8668	11.8309	**12.0725**
	Barbara	3679.06	3764.37	3538.31	12.4734	12.3738	12.6428
	School	5473.18	5575.17	5236.56	10.7484	10.6682	10.9403
	Desk	6717.85	6915.61	6507.47	9.8584	9.7324	9.9966
MC^*	Lena	5104.96	4740.12	3487.87	11.0508	11.3729	12.7051
	Barbara	3574.27	3664.64	4034.81	12.5989	12.4904	12.0726
	School	4552.19	5684.24	6595.23	11.5485	10.5840	9.9385
	Desk	6419.76	6585.28	6444.05	10.0556	9.94	10.0392
MCP^*	Lena	5163.93	4818.30	3589.89	11.0009	11.3018	12.5799
	Barbara	3799.47	3739.79	4179.64	12.3335	12.4023	11.9194
	School	4714.41	5725.38	6589.02	11.3965	10.5527	9.9425
	Desk	6554.94	6531.03	6460.64	9.9651	9.9809	10.0280
$MCPB^*$	Lena	**5518.97**	**4843.37**	**3693.77**	**10.7122**	**11.2793**	12.4560
	Barbara	**3857.33**	**3835.89**	**4353.05**	**12.2679**	**12.2921**	**11.7428**
	School	4971.37	5461.07	6661.51	11.1660	10.7580	9.8950
	Desk	6486.46	6479.79	6594.44	10.0107	10.0151	9.9390

4.2.3 Multi-Structural Similarity (MSSIM)

Multi-Structural Similarity compares the plaintext and the ciphertext. The SSIM values smaller than 0.1 indicate no correlation between the plaintext and cyphertext[21].

The SSIM algorithm can extract structural information from both plain

Table 4.2: Mean Square Root (MSE) and Peak signal-to-noise ratio (PSNR) - **Full Encryption without CBC**

		MSE			PSNR		
Process	Image	Red	Green	Blue	Red	Green	Blue
M^*	Lena	8191.31	8313.31	**7758.18**	8.99726	8.93306	**9.23320**
	Barbara	8123.18	8298.68	7822.12	9.03353	8.94070	9.19755
	School	8102.99	8338.57	7797.97	9.04434	8.91988	9.21098
	Desk	8116.72	8351.19	7877.87	9.0369	8.9133	9.1667
MC^*	Lena	9610.76	8928.82	6666.70	8.30322	8.62285	9.89169
	Barbara	8224.39	8298.29	8861.96	8.97976	8.94091	8.65550
	School	6256.39	8987.77	10354.80	10.16756	8.59428	7.97938
	Desk	7573.85	7910.56	7788.13	9.3376	9.1487	9.2164
MCP^*	Lena	9734.66	9041.25	6828.15	8.24759	8.56851	9.78777
	Barbara	8524.02	8438.18	9231.36	8.82435	8.86831	8.47814
	School	6483.12	9048.71	10354.93	10.01295	8.56493	7.97932
	Desk	7723.14	7810.31	7846.90	9.2528	9.20411	9.1838
$MCPB^*$	Lena	**10695.91**	**9099.53**	7029.11	**7.83862**	**8.54061**	9.66179
	Barbara	**8672.67**	**8738.30**	**9685.54**	**8.74927**	**8.71653**	**8.26956**
	School	7320.41	8609.96	10948.30	9.48544	8.78078	7.73733
	Desk	7717.01	7755.04	7904.35	9.2563	9.23495	9.15213

and encrypted images, enabling the identification of correlations between the two matrices.[21].

Our proposed work uses colour images, so the SSIM is multi-structural (MSSIM). This means it produces three matrices for each image: one for the Red channel, one for the Green channel, and the Blue channel.

Using CBC mode in our encryption scheme, it can be noticed that we achieved MSSIM values less than 0.1 for full encryption, which is the optimal value (table 4.5).

Meanwhile, whether using CBC mode or not, the values were higher than 0.1 in the selective encryption (table 4.5). This can be justified due to not encrypting all the images but only their selected essential blocks (region of interest). Even though these output values are not considered near the optimal, omitting the unencrypted blocks from the MSSIM calculation can achieve the optimal values, as shown for the fully encrypted images in table 4.6.

Table 4.3: Mean Square Root (MSE) and Peak signal-to-noise ratio (PSNR) - **Selective Encryption with CBC**

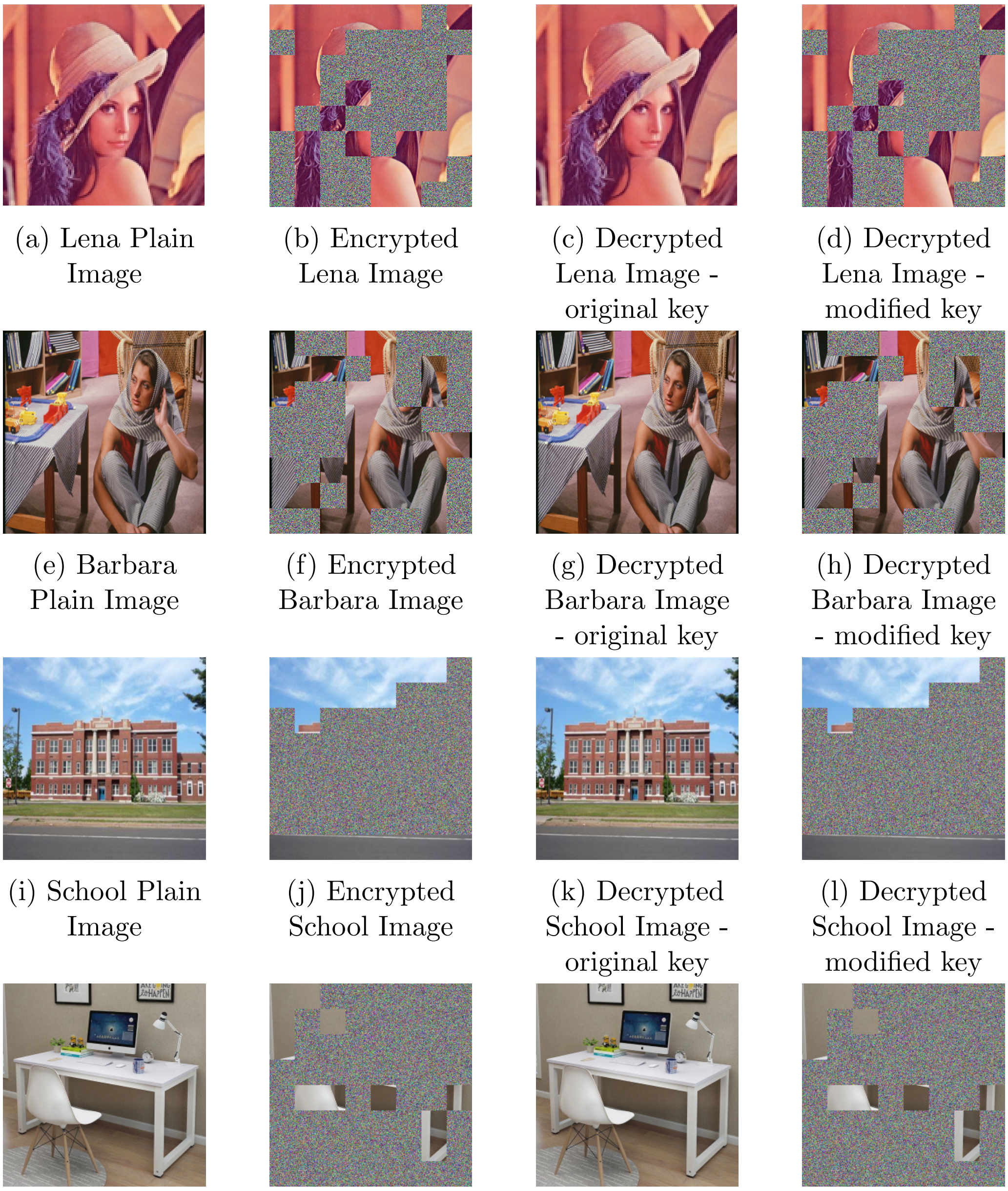
		MSE			PSNR		
Process	Image	Red	Green	Blue	Red	Green	Blue
M^*	Lena	5492.35	4814.80	3721.32	10.7332	11.3050	12.4238
	Barbara	3865.97	3829.64	4397.45	12.2582	12.2992	11.6987
	School	4989.15	5392.81	6725.37	11.1505	10.8126	9.8536
	Desk	6466.86	6424.25	6643.11	10.02386	10.05257	9.90708
MC^*	Lena	5492.35	4814.80	3721.32	10.7520	11.3069	12.4031
	Barbara	3865.97	3829.64	4397.45	12.2582	12.2992	11.6987
	School	4976.87	5396.07	6732.63	11.1612	10.8100	9.8489
	Desk	6451.41	6419.43	6652.05	10.03425	10.05583	9.90124
MCP^*	Lena	**5499.41**	**4835.13**	**3752.43**	**10.7276**	**11.2867**	**12.3876**
	Barbara	**3884.08**	3822.39	**4403.68**	**12.2379**	12.3074	**11.6926**
	School	4977.89	5398.27	6722.93	11.1603	10.8082	9.8552
	Desk	6421.80	6437.72	6660.22	10.05423	10.04348	9.8959
$MCPB^*$	Lena	5485.67	4817.33	3732.30	10.7385	11.3027	12.4110
	Barbara	3878.06	**3859.80**	4377.78	12.2446	**12.2651**	11.7182
	School	4990.22	5414.45	6714.34	11.1496	10.7952	9.8607
	Desk	6463.05	6415.70	6654.27	10.0264	10.0583	9.8997

4.2.4 Key Sensitivity Test

A slight change of initial parameters in the cryptosystem scheme should result in a different ciphertext image [14]. Changing a 1-bit key difference should result in a 50% difference of the ciphertexts between encrypting with the original key and the modified one; this ensures a high level of key sensitivity of the proposed SE method [21].In other words, a tiny modification in the private key results in an unrecognisable recovered image [23]. In our test, decrypting a cypher image with this 1-bit change resulted in a different plain image; the modified key could not restore the plain image as shown in figure 4.2.

4.2.5 Correlation Coefficient analysis (CC)

The range values of CC are between zero to one. Conducting values close to zero indicate secure encryption and no indication of a relationship between plain image and their cypher image [23]. Otherwise, values near 1 mean a



(a) Lena Plain Image

(b) Encrypted Lena Image

(c) Decrypted Lena Image - original key

(d) Decrypted Lena Image - modified key

(e) Barbara Plain Image

(f) Encrypted Barbara Image

(g) Decrypted Barbara Image - original key

(h) Decrypted Barbara Image - modified key

(i) School Plain Image

(j) Encrypted School Image

(k) Decrypted School Image - original key

(l) Decrypted School Image - modified key

(m) Desk Plain Image

(n) Encrypted Desk Image

(o) Decrypted Desk Image - original key

(p) Decrypted School Image - modified key

Figure 4.2: Key sensitivity Analysis Test Result

Table 4.4: Mean Square Root (MSE) and Peak signal-to-noise ratio (PSNR) - **Full Encryption with CBC**

		MSE			PSNR		
Process	Image	Red	Green	Blue	Red	Green	Blue
M^*	Lena	10626.80	9068.31	7077.21	7.86673	8.55553	9.63217
	Barbara	8628.58	8695.14	9816.76	8.77141	8.73803	8.21111
	School	7350.97	8462.80	11008.30	9.46735	8.85566	7.71360
	Desk	7677.69	7713.91	7932.39	9.2784	9.2580	9.1367
MC^*	Lena	10616.7	9052.98	7099.86	7.87089	8.56288	9.61830
	Barbara	8652.58	**8722.51**	9794.79	8.75934	**8.72438**	8.22085
	School	7317.84	8486.89	11042.70	9.48696	8.84331	7.70004
	Desk	7693.33	7679.63	7939.49	9.26965	9.27739	9.13287
MCP^*	Lena	**10695.6**	9022.60	**7131.03**	7.83872	8.57748	9.59927
	Barbara	8659.19	8703.29	9778.64	8.75602	8.73396	8.22801
	School	7322.91	8484.37	10979.31	9.48396	8.84460	7.72504
	Desk	7679.09	7667.16	7979.38	9.27770	9.28445	9.11110
$MCPB^*$	Lena	10695.1	**9075.04**	7100.13	7.83892	8.55231	9.61813
	Barbara	**8677.55**	8719.13	**9797.78**	**8.74682**	8.72607	**8.21952**
	School	7317.32	8495.68	11059.42	9.48727	8.83881	7.69347
	Desk	7693.77	7682.75	7982.96	9.26940	9.27563	9.10916
[23]	Lena (Periodic Periodic VCA)	82.38	82.56	93.37	28.95	28.95	28.42
[23]	Lena (Periodic Null VCA)	82.48	82.87	93.37	28.96	28.94	28.41

highly correlated ciphertext and fail to resist statistical attacks [14]. Table 4.7 illustrates correlation coefficient values for the plain images used in this thesis with values near one. The mathematical computation for correlation coefficient C.C. is as follows [6]:

$$C.C = \frac{\frac{1}{N} \times \sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\frac{1}{N} \times \sum_{i=1}^{N}(x_i - E(x))^2} \times \sqrt{\frac{1}{N} \times \sum_{i=1}^{N}(y_i - E(y))^2)}} \tag{4.6}$$

$$E(x) = \frac{1}{N} \times \sum_{i=1}^{N}(x_i \tag{4.7}$$

$$E(y) = \frac{1}{N} \times \sum_{i=1}^{N}(y_i \tag{4.8}$$

The above-mentioned equations involve variables such as N, which denotes the number of pixels. Additionally, x_i and y_i represent the values of neighbouring pixels in the plain and ciphered images.

Table 4.5: Multiscale structural Similarity (MSSIM) - **Selective Encryption**

		Without CBC			With CBC		
Process	Image	Red	Green	Blue	Red	Green	Blue
M^*	Lena	0.50646	0.53230	**0.43148**	**0.44589**	0.49008	0.49336
	Barbara	**0.55302**	0.56647	0.62371	0.56681	0.55806	**0.54319**
	School	0.31276	0.26902	0.36622	0.3672	0.3526	0.3334
	Desk	0.22819	0.20675	0.26054	0.22850	0.22802	0.22814
MC^*	Lena	0.45278	**0.48557**	0.51333	0.44847	0.49149	**0.49206**
	Barbara	0.59895	0.55207	0.56734	0.56591	0.55414	0.54439
	School	0.38154	0.32505	0.30815	0.3706	0.3558	0.3340
	Desk	0.22590	0.20659	0.21748	0.22850	0.22802	0.22814
MCP^*	Lena	0.44986	0.48637	0.48872	0.45276	**0.48876**	0.49209
	Barbara	0.56631	**0.54499**	0.54853	0.56181	0.55272	0.54656
	School	0.34446	0.31161	0.31850	0.3686	0.3549	0.3382
	Desk	0.21505	0.21602	0.21800	0.22850	0.22802	0.22814
$MCPB^*$	Lena	**0.44293**	0.48946	0.49960	0.45077	0.48912	0.49324
	Barbara	0.56223	0.55287	**0.54531**	**0.55917**	**0.54936**	0.54416
	School	0.37119	0.34939	0.34457	0.3612	0.3531	0.3339
	Desk	0.22198	0.22859	0.23242	0.22850	0.22802	0.22814

In our correlation coefficient analysis test, as shown in the tables 4.11 and 4.10, values for full encryption are very close to zero and much closer to the optimal value in comparison to related work in [18] and [23]. Meanwhile, selective encryption with CBC results has decreased the C.C. values to around 0.3, which is close to zero and acceptable to selective image encryption.

4.2.6 Information entropy analysis

To measure the average information in an image, entropy analysis is used [14]. For a true random image emitting 256 values (index i from 0 to 255), an 8 bits value is the ideal entropy value [14] [6] [23]. The ideal value indicates a cryptosystem scheme is resistant to entropy attack. The mathematical equation for information entropy H of an image C is as follows [14] [6]:

$$H = \sum_{i=0}^{N-1} Pro(c_i) \times log_2 \frac{1}{Proc(c_i)} \tag{4.9}$$

Table 4.6: Multiscale structural Similarity (MSSIM) - **Full Encryption**

		Without CBC			With CBC		
Process	Image	Red	Green	Blue	Red	Green	Blue
M^*	Lena	0.19236	0.15795	**0.06298**	0.08011	0.06973	0.09419
	Barbara	0.08131	0.14222	0.22913	0.05502	0.06086	**0.05436**
	School	0.07940	0.03706	0.11357	0.07289	0.08212	0.08050
	Desk	0.10352	0	0.1161	0.08123	0.07955	0.08262
MC^*	Lena	0.08048	**0.06436**	0.11549	0.07949	0.06841	0.09279
	Barbara	0.10982	0.07775	0.12578	0.05686	0.06269	0.05726
	School	0.11312	0.06887	0.05065	0.07462	0.07965	0.07722
	Desk	0.08418	0.07123	0.08435	0.08123	0.07955	0.08262
MCP^*	Lena	**0.07036**	0.07570	0.06919	0.07764	0.07273	**0.08816**
	Barbara	0.06677	**0.05018**	0.06938	**0.05262**	0.06006	0.05464
	School	0.06075	0.05496	0.06291	0.07180	0.07748	0.08270
	Desk	0.07165	0.07769	0.07683	0.08123	0.07955	0.08262
$MCPB^*$	Lena	0.07237	0.07218	0.09266	**0.07495**	**0.06756**	0.09216
	Barbara	**0.05051**	0.05574	**0.06292**	0.05456	**0.05833**	0.05905
	School	0.07770	0.07848	0.08562	0.07236	0.08064	0.07723
	Desk	0.07869	0.08306	0.08326	0.08123	0.07955	0.08262

Table 4.7: Coefficient Correlation (CC) - **Plain Images**

	Diagonal			Horizontal			Vertical		
Image	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
Lena	0.96969	0.95554	0.91828	0.97977	0.96906	0.93274	0.98931	0.98249	0.95760
Barbara	0.86324	0.84339	0.86502	0.87918	0.85967	0.88150	0.95437	0.95025	0.95636
School	0.90859	0.94185	0.96743	0.94575	0.96595	0.98109	0.96027	0.97423	0.9853
Desk	0.92722	0.93170	0.93941	0.95531	0.95787	0.96301	0.95134	0.95486	0.95994

Where N denotes the total number of pixels, which is 256-pixel values, and $Pro(c_i)$, is the probability of occurrence of each value ($i = 0, 1, \ldots., 255$). The maximal information entropy for 256 symbols is 8 bits such that $Pro(c_i) = 2^{-8}$, and

$$H = \sum_{i=0}^{256-1} 2^{-8} \times log_2 \frac{1}{2^{-8}} \tag{4.10}$$

$$H = 8 \tag{4.11}$$

Tables 4.13 and 4.14 in our information entropy analysis display values

Table 4.8: Coefficient Correlation (CC) - **Selective Encryption without CBC Mode**

		Diagonal			Horizontal			Vertical		
Process	Image	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
M^*	Lena	0.42465	0.42236	0.27553	0.38056	0.46089	0.33111	0.44201	0.44270	0.28409
	Barbara	0.42032	0.41140	0.43720	0.41694	0.45380	0.48560	0.49011	0.48111	0.49835
	School	0.27601	0.31720	0.42390	0.27205	0.35487	0.46536	0.29858	0.34052	0.43573
	Desk	0.22978	0.24466	0.22451	0.20499	0.24066	0.24428	0.23728	0.25617	0.22764
MC^*	Lena	0.42465	0.42236	0.27553	0.38056	0.46089	0.33111	0.44201	0.44270	0.28409
	Barbara	0.39666	0.36598	0.41581	0.39194	0.40147	0.45880	0.45542	0.42756	0.47401
	School	0.19481	0.32223	0.42102	0.19396	0.36152	0.46877	0.21349	0.34730	0.43200
	Desk	0.15780	0.17176	0.17102	0.13916	0.17557	0.19884	0.17279	0.18464	0.17688
MCP^*	Lena	**0.37712**	0.36376	0.16601	0.38057	0.39045	0.17871	0.41916	0.42735	0.23150
	Barbara	0.36906	0.32826	0.38277	0.37884	0.35302	0.39919	0.45527	0.42775	0.47372
	School	0.16246	0.27758	0.38711	0.17375	0.31740	0.41048	0.21338	0.34724	0.43197
	Desk	0.11965	0.12011	0.12001	0.12462	0.15701	0.13819	0.17280	0.18482	0.17672
$MCPB^*$	Lena	0.38441	**0.31237**	**0.14867**	0.37593	**0.32018**	**0.14867**	**0.39362**	**0.31647**	**0.15755**
	Barbara	**0.35495**	**0.32747**	**0.37593**	**0.36127**	**0.33176**	**0.37593**	**0.40440**	**0.36336**	**0.42132**
	School	0.13071	0.23527	0.36554	0.11747	0.24761	0.36554	0.13367	0.22265	0.36372
	Desk	0.05954	0.06372	0.06863	0.05869	0.06960	0.06863	0.06864	0.04321	0.07370

Table 4.9: Coefficient Correlation (CC) - **Selective Encryption with CBC Mode**

		Diagonal			Horizontal			Vertical		
Process	Image	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
M^*	Lena	0.37549	0.31376	**0.15298**	**0.38166**	0.31924	0.15979	0.38467	0.32438	0.15900
	Barbara	0.35655	0.32632	0.37779	0.35885	0.32853	0.38388	**0.39937**	**0.36811**	**0.42115**
	school	0.1205	0.2326	0.3660	0.1262	0.2387	0.3697	0.1233	0.2340	0.3673
	Desk	0.05591	0.06160	0.07420	0.05761	0.06467	0.07076	0.05612	0.06720	0.07054
MC^*	Lena	0.37735	**0.30911**	0.15403	0.38426	**0.31737**	**0.15676**	0.38819	**0.32310**	0.16117
	Barbara	0.35720	0.32899	**0.37768**	0.35932	0.32982	0.38365	0.40051	0.36999	0.42181
	school	0.1216	0.2344	0.3635	0.1247	0.2375	0.3670	0.1247	0.2343	0.3658
	Desk	0.05921	0.05871	0.07375	0.05817	0.06243	0.07084	0.05659	0.06745	0.07364
MCP^*	Lena	**0.37479**	0.31453	0.15535	0.38302	0.32171	0.15981	**0.38161**	0.32548	0.16386
	Barbara	**0.35379**	**0.32523**	0.37777	**0.35633**	**0.32684**	0.38482	0.40133	0.37095	0.42209
	school	0.1227	0.2338	0.3615	0.1270	0.2368	0.3675	0.1197	0.2340	0.3621
	Desk	0.05880	0.06465	0.06986	0.05581	0.06088	0.07187	0.06012	0.06480	0.07281
$MCPB^*$	Lena	0.38029	0.31187	0.15851	0.38485	0.32043	0.15851	0.38790	0.32449	0.16156
	Barbara	0.35965	0.32681	0.37921	0.35868	0.32762	**0.37921**	0.40000	0.37037	0.42198
	school	0.1216	0.2344	0.3635	0.1247	0.2375	0.3670	0.1247	0.2343	0.3658
	Desk	0.05566	0.06067	0.06682	0.05501	0.06438	0.06682	0.05648	0.06604	0.07288

that are close to the ideal IE value of 8.0.

4.2.7 Histogram Analysis

Statistical analysis is a prevalent method for cryptosystem attacks. For a cryptosystem to be deemed strong against these attacks, the histogram of the encrypted image must exhibit uniform distribution. An Ideal cryptosystem of the ciphertext has a flat histogram significantly different from the original image. However, this uniform distribution of pixels reveals no significant information about the ciphertext statistics[5].

Table 4.10: Coefficient Correlation (CC) - **Full Encryption without CBC Mode**

		Diagonal			Horizontal			Vertical		
Process	Image	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
M^*	Lena	0.11851	0.18479	0.10940	0.03258	0.23714	0.18801	0.13433	0.20137	0.10887
	Barbara	0.09644	0.11652	0.12135	0.09265	0.17919	0.19244	0.12698	0.15507	0.14631
	school	0.18816	0.127428	0.13295	0.18098	0.15033	0.164778	0.20675	0.14993	0.14453
MC^*	Lena	0.07200	0.08785	0.09473	-0.0118	0.14500	0.17635	0.08916	0.11326	0.10560
	Barbara	0.06001	0.07056	0.07995	0.05700	0.12950	0.14630	0.07903	0.09888	0.10158
	school	0.11268	0.13415	0.10117	0.10952	0.16206	0.14341	0.13089	0.16058	0.11412
MCP^*	Lena	0.02799	0.02660	0.02912	0.02667	0.06656	0.04068	0.08916	0.11326	0.10560
	Barbara	0.01265	0.01586	0.02451	0.02445	0.04789	0.03813	0.07903	0.09888	0.10158
	school	0.07613	0.06221	0.04775	0.083830	0.11341	0.07981	0.13089	0.16058	0.11412
$MCPB^*$	Lena	**0.01124**	**0.00382**	**-0.0052**	**-0.0159**	**0.00296**	**-0.0052**	**0.01592**	**-0.0111**	**-0.0036**
	Barbara	**-0.0007**	**0.00186**	**0.00057**	**-0.0018**	**0.00653**	**0.00057**	**0.00706**	**-0.0130**	**-0.0004**
	school	0.01118	0.001386	-0.00507	-0.00852	0.01394	-0.00507	0.016468	-0.02286	-0.007571

Table 4.11: Coefficient Correlation (CC) - **Full Encryption with CBC Mode**

		Diagonal			Horizontal			Vertical		
Process	Image	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
M^*	Lena	0.00273	0.00152	-0.0009	**-0.0015**	**-0.0003**	-0.0015	0.00037	**-0.0026**	0.00079
	Barbara	-0.0017	0.00022	-0.0002	-0.0011	0.00065	0.00167	**-0.0012**	0.00301	-0.0021
	School	-0.00103	-0.00193	-0.00194	-0.00146	0.00047	0.00098	-0.001862	0.00060	0.00273
	Desk	0.18194	0.19839	0.16374	0.14864	0.19034	0.18765	0.18736	0.21081	0.16521
MC^*	Lena	0.00174	-0.0007	**-0.0041**	0.00030	0.00276	**-0.0035**	0.00388	-0.0015	0.00148
	Barbara	0.00051	**-0.0013**	4.80169	-2.5043	0.00084	0.00137	-0.0011	0.00144	0.00265
	School	0.00339	-0.00175	-0.00145	-0.00057	0.00011	0.001192	0.00084	0.000004	0.00024
	Desk	0.11207	0.12256	0.12357	0.08662	0.12163	0.15356	0.12819	0.13662	0.12783
MCP^*	Lena	**-0.0014**	-0.0011	9.92108	-0.0013	0.00051	-0.0004	**-0.0008**	0.00464	**-0.0025**
	Barbara	**-0.0019**	0.00072	-0.0009	0.00223	**-0.0044**	-7.7609	0.00468	0.00442	0.00145
	School	0.00082	-0.00036	0.00198	0.00212	0.00358	0.00118	0.00076	0.00107	-0.00251
	Desk	0.06902	0.06869	0.06383	0.07197	0.10589	0.08155	0.12819	0.13662	0.12783
$MCPB^*$	Lena	-0.0013	**-0.0032**	-0.0013	0.00032	0.00404	-0.0013	0.00252	0.00294	0.00469
	Barbara	-0.0006	0.00236	**-0.0011**	**-0.0026**	5.06528	**-0.0011**	**0.00255**	**-0.0004**	**-0.0022**
	School	0.00057	-0.00224	0.003610	-0.00148	-0.00085	0.00361	0.00015	-0.00042	-0.002480
	Desk	0.00493	0.00086	-0.00347	-0.00271	0.00564	-0.00347	0.01272	-0.02249	-0.00111
[18]	Lena	0.0062	0.0067	0.0044	-0.0075	-0.0050	-0.0035	0.0004	-0.0018	0.0026
[18]	Barbara	-0.0029	-0.0003	-0.0009	0.0013	0.0008	-0.0003	-0.0014	-0.0004	0.0007
[23]	Lena (Periodic VCA)		0.0010			0.0030			-0.0011	
[23]	Lena (Null VCA)		0.0053			0.0078			-0.0042	

Table 4.15 illustrates the histogram graph for the plain images used. It is noticed that the distribution is not flat. After performing the full encryption over Lena and Barbara's images, the histogram diagrams in tables 4.17 and 4.19 become flatter once the algorithm has implemented all its stages performing exclusive-or over the images' bytes.

4.2.8 Plain-text sensitivity attack

To prevent known and chosen plaintext attacks on a cryptosystem, it should be designed to detect even the slightest alteration in the plain text. This is crucial for ensuring the system's security, as attackers may use multiple plain-

Table 4.12: Information Entropy (IE) - **Plain Images**

Image	Red	Green	Blue
Lena	7.25310	7.59403	6.96842
Barbara	7.25310	7.59403	6.96842
School	7.34996	7.47045	7.36521
Desk	7.41893	7.50148	7.57541

Table 4.13: Information Entropy (IE) - **Selective Encryption**

		Without CBC			With CBC		
Process	Image	Red	Green	Blue	Red	Green	Blue
M^*	Lena	7.56818	7.86548	7.62250	7.82460	7.89268	7.73739
	Barbara	7.90473	7.84312	7.80775	7.92790	7.85791	7.86443
	School	7.84903	7.80466	7.46922	7.8960	7.8788	7.7796
	Desk	7.88203	7.93067	7.91125	7.96877	7.96844	7.9681
MC^*	Lena	7.76804	7.80864	7.70359	7.82536	7.89226	7.73808
	Barbara	7.89245	7.84333	7.84253	7.92773	7.85755	7.86542
	School	7.78031	7.87580	7.68255	7.8962	7.8788	7.7780
	Desk	7.93312	7.89921	7.92537	7.96877	7.96844	7.96819
MCP^*	Lena	7.76804	7.80864	7.70359	7.82721	7.89324	**7.73931**
	Barbara	7.89245	7.84333	7.84253	7.92736	7.85754	**7.86695**
	School	7.78031	7.87580	7.68255	7.8961	7.8797	7.7772
	Desk	7.93312	7.89921	7.92537	7.96877	7.96844	7.96819
$MCPB^*$	Lena	**7.82752**	**7.89408**	**7.73775**	**7.82844**	**7.89359**	7.73822
	Barbara	**7.92657**	**7.85652**	**7.86324**	**7.92815**	**7.85894**	7.86551
	School	7.89542	7.87903	7.77997	7.8967	7.8811	7.7782
	Desk	7.96830	7.96931	7.96793	7.96877	7.96844	7.96819

text samples with minor differences to analyse the corresponding ciphertexts. A specific procedure tests a cryptosystem's sensitivity to such attacks. This involves measuring the system's ability to detect one-bit changes in the plain text and analyzing the differences between the corresponding ciphertexts of multiple plain-text samples. Several studies, such as those by [17, 19], have emphasised the importance of this requirement in building a secure cryptosystem. The procedure for evaluating a cryptosystem's vulnerability to such attacks involves the following steps:

1. Choose the initial plain image as P_1.

2. Create a new variable P_2 by altering one bit in P_1. Specifically, the two variables should be identical except for one bit. This bit can be

Table 4.14: Information Entropy (IE) - **Full Encryption**

		Without CBC			With CBC		
Process	Image	Red	Green	Blue	Red	Green	Blue
M^*	Lena	7.68036	7.96525	7.81668	7.99939	7.99922	7.99924
	Barbara	7.95441	7.95797	7.90873	**7.99931**	7.99932	7.99928
	School	7.94474	7.85278	7.48580	7.99918	7.99921	7.99917
	Desk	7.90302	7.94873	7.93251	7.99932	7.99933	7.99917
MC^*	Lena	7.83589	7.71196	7.96696	7.99921	7.99930	**7.99935**
	Barbara	7.91825	7.95383	7.97790	7.99921	7.99936	7.99937
	School	7.54978	7.93114	7.83032	7.99934	7.99923	7.99930
	Desk	7.93904	7.90982	7.95030	7.99932	7.99933	7.99917
MCP^*	Lena	7.83589	7.71196	7.96696	**7.99941**	7.99927	7.99931
	Barbara	7.91825	7.95383	7.97790	7.99916	7.99933	**7.99939**
	School	7.54978	7.93114	7.83032	7.99916	7.99929	7.99915
	Desk	7.93904	7.90982	7.95030	7.99932	7.99933	7.99917
$MCPB^*$	Lena	**7.99923**	**7.99925**	**7.99906**	7.99935	**7.99934**	7.99934
	Barbara	**7.99933**	**7.99934**	**7.99921**	7.99923	**7.99936**	7.99931
	School	7.99909	7.99897	7.99871	7.99932	7.99939	7.99920
	Desk	7.99924	7.999181	7.99926	7.99932	7.99933	7.99917

located at the first block's beginning, middle, or end. The plaintext results should be averaged across these three scenarios.

3. The secret key used to encrypt both images (P_1 and P_2) is the same.

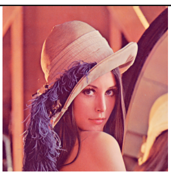
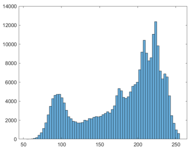
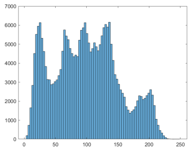
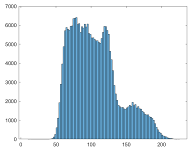
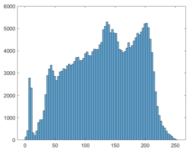
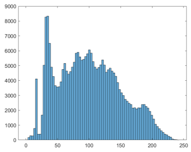
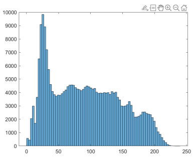
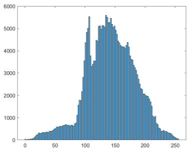
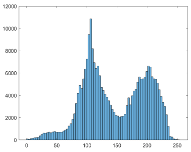
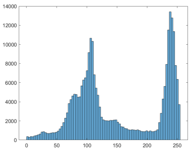
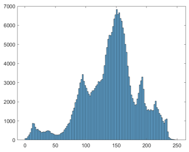
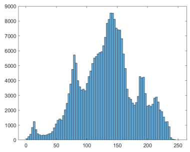
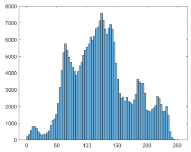
4. Two cypher images, namely C_1 and C_2, are generated by the encryption process that was previously executed.

5. The encrypted images C_1 and C_2 undergo a series of statistical security tests.

To gauge a cryptosystem's ability to withstand plain-text sensitivity attacks, most researchers rely on two security parameters: the Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). These parameters are calculated using the following equations, respectively:

$$NPCR = \frac{1}{L \times C \times P} \times \sum_{p=1}^{P} \sum_{i=1}^{L} \sum_{j=1}^{C} D(i, j, p) \times 100\% \tag{4.12}$$

Table 4.15: Original Images Histogram

Image	Red	Green	Blue
			
			
			
			

where

$$D(i,j,p) = \begin{cases} 0, & \text{if} C_1(i,j,p) = C_2(i,j,p) \\ 1, & \text{if} C_1(i,j,p) \neq C_2(i,j,p) \end{cases} \tag{4.13}$$

$$\text{UACI} = \frac{1}{L \times C \times P \times 255} \times \sum_{p=1}^{P} \sum_{i=1}^{L} \sum_{j=1}^{C} D(i,j,p) |C_1(i,j,p) - C_2(i,j,p)| \times 100\% \tag{4.14}$$

In the equations presented earlier, the image's row, column, and plane indexes are denoted by i, j, and p. The image's length, width, and plane

Table 4.16: Histogram Lena Image: **Selective Encryption**

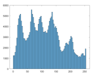
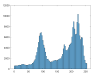
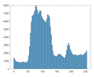
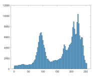
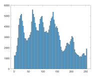
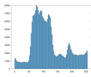
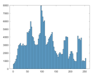
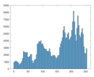
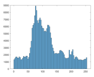
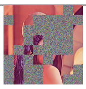
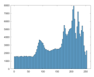
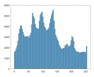
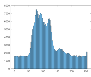
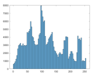
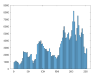
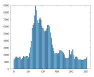
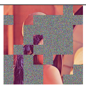
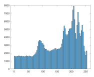
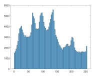
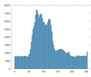
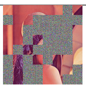
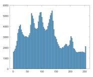
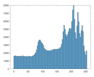
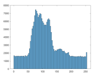
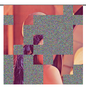
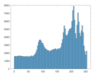
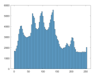
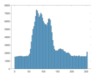
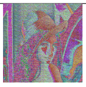
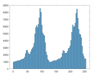
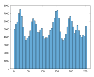
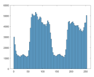
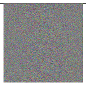
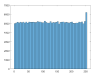
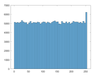
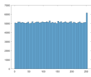
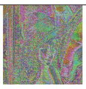
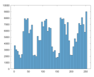
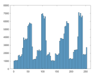
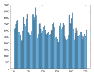
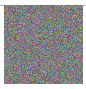
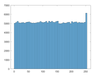
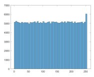
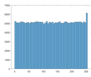
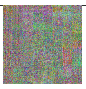
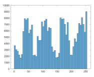
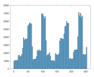
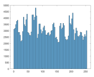
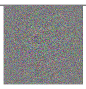
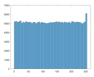
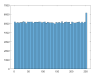
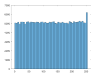
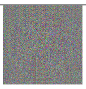
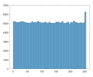
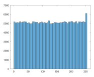
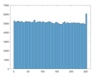
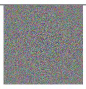
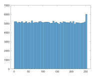
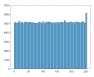
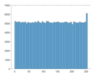
	Without CBC				With CBC			
Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
M								
MC								
MCP								
MCPB								

Table 4.17: Histogram Lena Image: **Full Encryption**

	Without CBC				With CBC			
Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
M								
MC								
MCP								
MCPB								

sizes are represented by L, C, and P, respectively.

To assess a cryptosystem's resistance against differential attacks by Eli Biham and Adi Shamir, researchers use the metrics of NPCR and UACI. The ideal values for these metrics are 99.61% and 33.46%, respectively. In our thesis, implementing this analysis over full encryption with CBC mode resulted in obtaining near optimal values as the following: NPCR score of 99.60% and UACI score: of 33.50%.

4.2.9 Time Performance

To measure the time efficiency of a cryptosystem, we must first analyze the complexity of its algorithm in terms of logical and mathematical operations

Table 4.18: Histogram Barbara Image: **Selective Encryption**

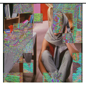
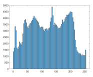
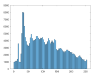
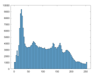
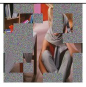
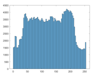
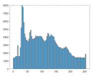
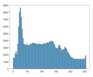
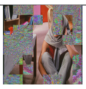
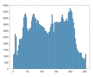
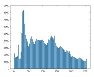
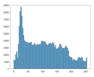
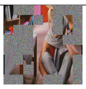
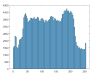
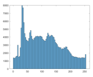
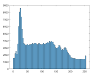
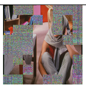
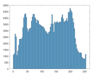
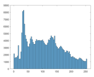
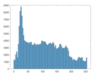
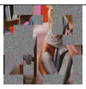
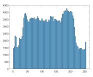
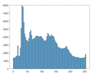
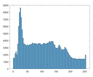
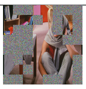
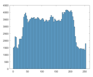
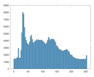
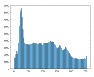
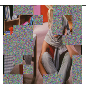
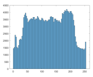
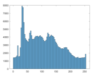
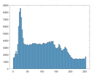
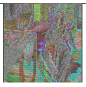
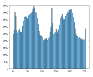
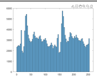
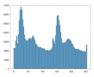
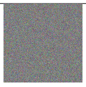
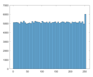
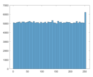
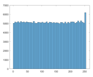
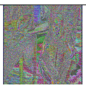
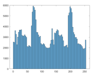
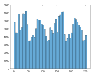
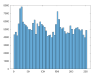
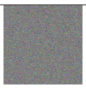
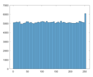
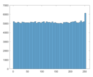
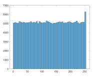
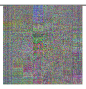
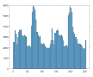
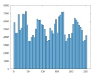
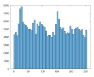
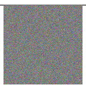
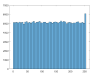
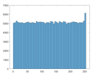
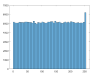
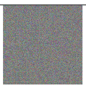
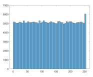
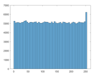
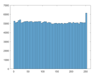
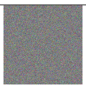
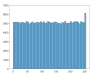
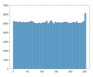
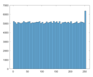
	Without CBC				With CBC			
Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
M								
MC								
MCP								
MCPB								

Table 4.19: Histogram Barbara Image: **Full Encryption**

	Without CBC				With CBC			
Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
M								
MC								
MCP								
MCPB								

and read-write memory operations. Next, we can determine its performance by evaluating its running speed, which can be measured through the average times it takes to encrypt or decrypt data, its encryption throughput, and the number of cycles required to encrypt one byte. The encryption throughput (ET) and number of cycles necessary for encryption or decryption of one byte are defined as:

$$ET = \frac{Image_{size(Byte)}}{Encryption_{Time}(Second)} \tag{4.15}$$

Table 4.20: Histogram School Image: **Selective Encryption**

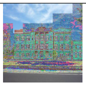
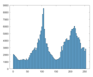
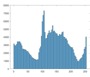
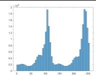
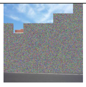
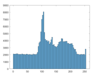
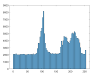
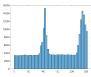
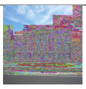
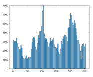
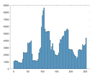
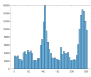
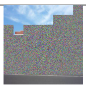
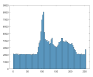
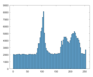
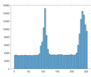
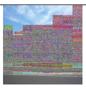
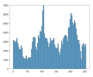
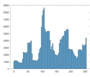
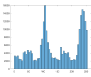
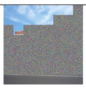
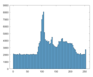
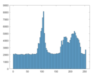
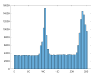
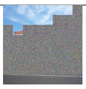
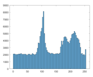
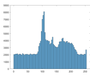
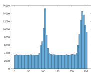
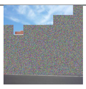
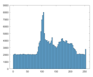
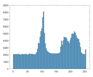
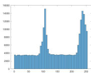
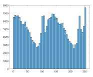
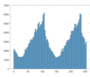
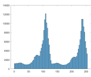
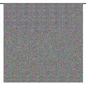
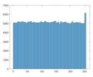
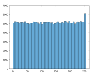
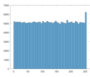
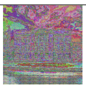
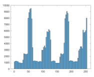
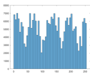
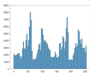
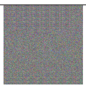
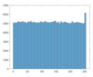
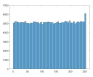
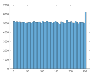
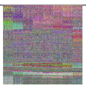
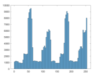
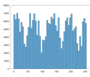
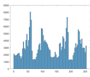
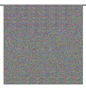
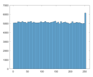
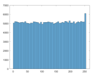
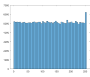
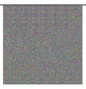
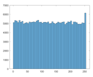
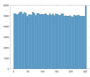
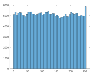
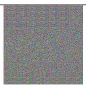
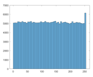
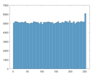
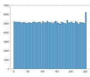
	Without CBC				With CBC			
Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
M								
MC								
MCP								
MCPB								

Table 4.21: Histogram School Image: **Full Encryption**

	Without CBC				With CBC			
Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
M								
MC								
MCP								
MCPB								

$$\text{Number of cycles per Byte} = \frac{\text{CPU Speed}_{(Hertz)}}{\text{ET}_{(Byte)}} \tag{4.16}$$

With the final equation, we can compare the operational speed of various cryptosystems running on different platforms. Table 4.24 demonstrates our time performance result. A Windows 10 PC was used to produce all the results with a 3.0 GHz processor and 32GB of RAM. The operating system was running on a 64-bit platform.

Table 4.22: Histogram Desk Image: **Selective Encryption**

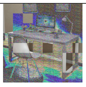
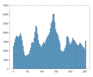
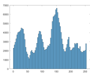
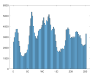
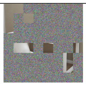
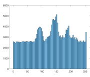
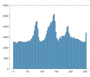
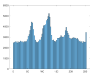
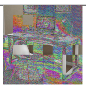
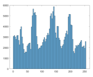
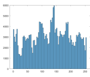
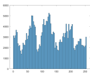
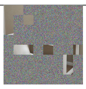
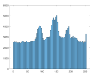
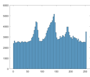
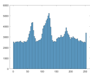
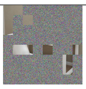
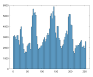
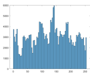
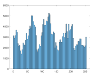
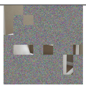
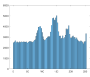
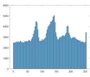
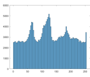
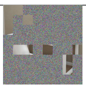
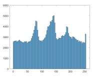
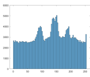
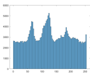
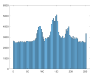
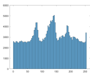
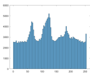
	Without CBC				With CBC			
Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
M								
MC								
MCP								
MCPB								

Table 4.23: Histogram Desk Image: **Full Encryption**

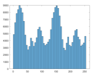
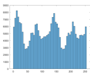
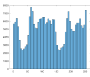
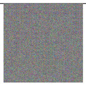
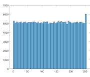
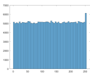
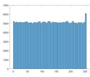
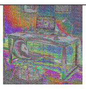
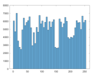
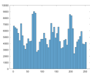
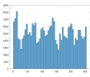
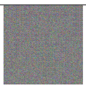
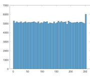
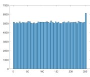
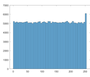
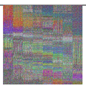
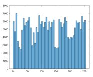
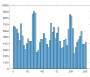
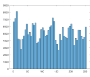
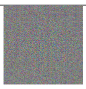
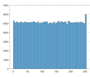
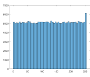
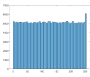
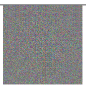
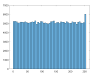
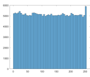
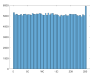
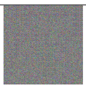
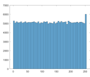
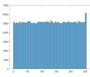
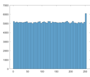
	Without CBC				With CBC			
Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
M								
MC								
MCP								
MCPB								

Table 4.24: Time Performance

Image	Encryption type	Size	Time (sec)	ET (MBps)	Cycles per byte
Lena	Full encryption	512x512	3.0620	47.78	159.27
Lena	Selective encryption	512x512	1.6460	25.68	85.62
Barbara	Full encryption	512x512	3.3620	40.41	134.72
Barbara	Selective encryption	512x512	1.9460	23.39	77.98
School	Full encryption	512x512	3.9220	36.83	114.72
School	Selective encryption	512x512	2.0460	20.65	57.98

Chapter 5

Conclusion and Future Work

5.1 Conclusion

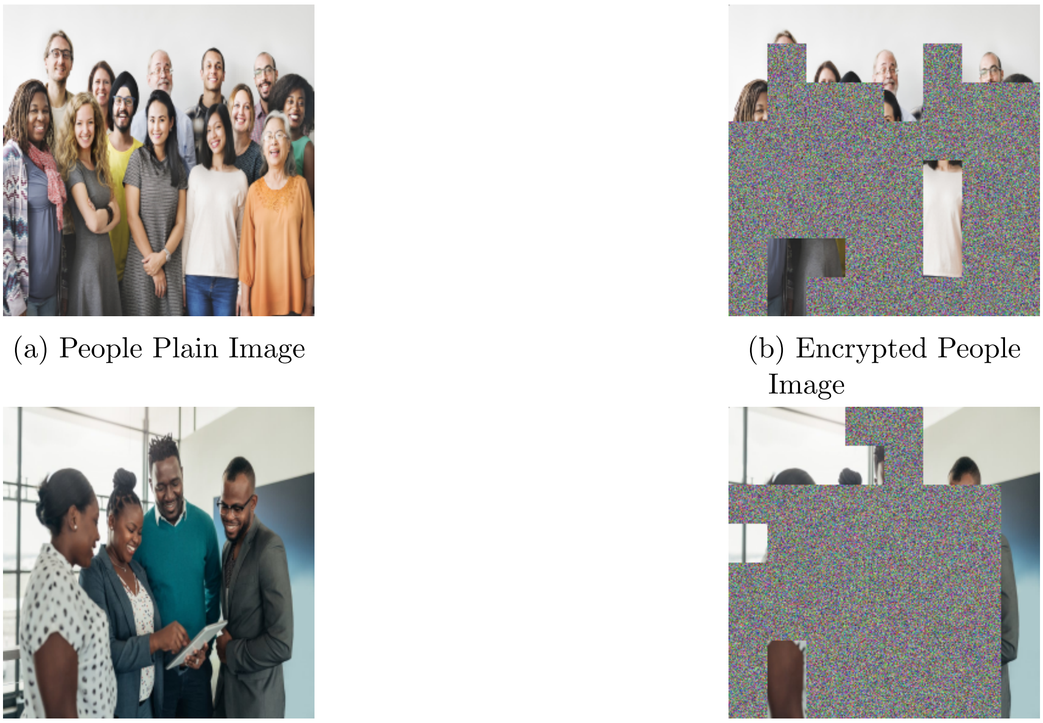
Secure data sharing on devices with limited resources requires efficient encryption algorithms that balance energy and memory usage to optimise device performance. However, traditional algorithms can be computationally intensive and lead to high communication overhead, which is ineffective for protecting data on limited resource devices. This is due to the pixels' properties being very correlated with their adjutant ones. In this thesis, an effective selective encryption algorithm has been developed for images. The algorithm divides the image into blocks and identifies which blocks require encryption. The most critical bit (8^{th} bit) on all bytes of the selected blocks is exclusive-xored. The other three most significant bits (7^{th}, 6^{th} and 5^{th} bit) between the red, green, and blue channels are swapped. The Fisher-Yates algorithm is utilised to scramble the pixels row-wise. An exclusive-or operation is performed on all bytes of the essential blocks. Each byte is XORed with the previous one, and a random byte is added. The algorithm is designed to adapt Cipher Block Chaining (CBC) Mode, ensuring robust security in image encryption.

The evaluation metrics show near-optimal values for the proposed encryption algorithm with an O(n) complexity. The average PSNR for selective encryption with CBC is 10.06, while MSSIM and IE are 0.4 and 7.8, respectively. A key sensitivity test shows a tiny modification in the private key results in an unrecognizable recovered image. We also conducted an average coefficient correlation of 0.224, NPCR score of 99.60%, and UACI score of 33.50%.

5.2 Future Work

Due to time limitations, this thesis used an online pseudo-random number generator, and as future work, researchers can develop a PRNG generator. Besides, our algorithm is adapted for coloured images; thus, it is possible to adapt changes to the proposed algorithm to include grey-scale images. One common approach that some researchers consider is the removal of pixel redundancy in images; in other words, compressing the image before the encryption. This can achieve lower time and computational cost of the algorithm. Moreover, the blocks were divided into equal sizes and scrambled column-wise in our proposed algorithm with the full encryption approach. Researchers can develop our proposed algorithm to divide the image into unequal block sizes and perform rotations over these blocks before performing the other encryption process. In addition, scramble the pixels column and row-wise involving the RGB channels.

After testing our algorithm with more complex images, such as a group of people (as illustrated in Figure 5.1), we noticed that the algorithm's output has limitations. Not all faces were encrypted, and other parts of the body were not encrypted, such as the arm of the woman. This observation could open up a new field of research and development for our algorithm. Artificial intelligence could potentially help identify regions of interest more dynamically and effectively.



(a) People Plain Image

(b) Encrypted People Image

(c) People Plain Image

(d) Encrypted People Image

Figure 5.1: Other test images

References

[1] Image processing standard test images. `http://www.eecs.qmul.ac.uk/~phao/IP/Labs/cwk/images/`. Accessed: 2023-08-21.

[2] Image processing standard test images. `https://testimages.juliaimages.org/stable/imagelist/`. Accessed: 2023-08-21.

[3] Random number generator ”stat trek. `https://stattrek.com/statistics/random-number-generator#table/`. Accessed: 2023-09-16.

[4] Jawad Ahmad and Fawad Ahmed. Efficiency analysis and security evaluation of image encryption schemes. *computing*, 23:25, 2010.

[5] Guanrong Chen, Yaobin Mao, and Charles K Chui. A symmetric image encryption scheme based on 3d chaotic cat maps. *Chaos, Solitons & Fractals*, 21(3):749–761, 2004.

[6] Mousa Farajallah. *Chaos-based crypto and joint crypto-compression systems for images and videos*. PhD thesis, Universite de Nantes, 2015.

[7] Mousa Farajallah, Guillaume Gautier, Wassim Hamidouche, Olivier Déforges, and Safwan El Assad. Selective encryption of the versatile video coding standard. *IEEE Access*, 10:21821–21835, 2022.

[8] Zeinab Fawaz, Safwan El Assad, M Frajallah, Ayman Khalil, René Lozi, and Olivier Déforges. Lightweight chaos-based cryptosystem for secure images. In *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*, pages 26–30. IEEE, 2013.

[9] Borko Furht. *Multimedia tools and applications*, volume 359. Springer Science & Business Media, 2012.

[10] S Geetha, P Punithavathi, A Magnus Infanteena, and S Siva Sivatha Sindhu. A literature review on image encryption techniques. *International Journal of Information Security and Privacy (IJISP)*, 12(3):42–83, 2018.

[11] Iyad Hraini, Mousa Farajallah, Nabil Arman, and Wassim Hamidouche. Joint crypto-compression based on selective encryption for wmsns. *IEEE Access*, 9:161269–161282, 2021.

[12] ISSA JACAMAN and MOUSA FARAJALLAH. A lightweight spatial domain image encryption algorithms: A review paper. *Journal of Theoretical and Applied Information Technology*, 101(3), 2023.

[13] Omed Khalind and Benjamin Aziz. Single-mismatch 2lsb embedding steganography. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 000283–000286. IEEE, 2013.

[14] Jan Sher Khan and Jawad Ahmad. Chaos based efficient selective image encryption. *Multidimensional Systems and Signal Processing*, 30:943–961, 2019.

[15] Manju Khari, Aditya Kumar Garg, Amir H Gandomi, Rashmi Gupta, Rizwan Patan, and Balamurugan Balusamy. Securing data in internet of things (iot) using cryptography and steganography techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(1):73–80, 2019.

[16] Osama A Khashan and Muath AlShaikh. Edge-based lightweight selective encryption scheme for digital medical images. *Multimedia Tools and Applications*, 79(35-36):26369–26388, 2020.

[17] Shiguo Lian, Jinsheng Sun, and Zhiquan Wang. Security analysis of a chaos-based image encryption algorithm. *Physica A: Statistical Mechanics and its Applications*, 351(2-4):645–661, 2005.

[18] Kaiyun Ma, Lin Teng, Xingyuan Wang, and Juan Meng. Color image encryption scheme based on the combination of the fisher-yates scrambling algorithm and chaos theory. *Multimedia Tools and Applications*, 80:24737–24757, 2021.

[19] Ismail Mansour, Gerard Chalhoub, and Bassem Bakhache. Evaluation of a fast symmetric cryptographic algorithm based on the chaos theory for wireless sensor networks. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 913–919. IEEE, 2012.

[20] Ayoub Massoudi, Frédéric Lefebvre, Christophe De Vleeschouwer, Benoit Macq, and J-J Quisquater. Overview on selective encryption of image and video: challenges and perspectives. *Eurasip Journal on information security*, 2008(1):179290, 2008.

[21] Han Qiu, Meikang Qiu, Meiqin Liu, and Zhong Ming. Lightweight selective encryption for social data protection based on ebcot coding. *IEEE Transactions on Computational Social Systems*, 7(1):205–214, 2019.

[22] Rawan Qumsieh, Mousa Farajallah, and Rushdi Hamamreh. Joint block and stream cipher based on a modified skew tent map. *Multimedia Tools and Applications*, 78:33527–33547, 2019.

[23] Satyabrata Roy, Manu Shrivastava, Chirag Vinodkumar Pandey, Sanjeet Kumar Nayak, and Umashankar Rawat. Ievca: An efficient image encryption technique for iot applications using 2-d von-neumann cellular automata. *Multimedia Tools and Applications*, 80:31529–31567, 2021.

[24] Tao Xiang, Jia Hu, and Jianglin Sun. Outsourcing chaotic selective image encryption to the cloud with steganography. *Digital Signal Processing*, 43:28–37, 2015.

[25] Erdem Yavuz. A novel chaotic image encryption algorithm based on content-sensitive dynamic function switching scheme. *Optics & Laser Technology*, 114:224–239, 2019.

[26] Yushu Zhang, Di Xiao, Wenying Wen, and Yuan Tian. Edge-based lightweight image encryption using chaos-based reversible hidden transform and multiple-order discrete fractional cosine transform. *Optics & Laser Technology*, 54:1–6, 2013.

[27] Djemel Ziou, Salvatore Tabbone, et al. Edge detection techniques-an overview. *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, 8:537–559, 1998.