Grenze International Journal of Engineering and Technology, June Issue

S-Box based Nibble-Swapping Encryption Scheme to Enhance the Security of Digital Images

Aishwarya N
Assistant Professor, Department of E&C Engineering, Dayananda Sagar College of Engineering, Bangalore, India
Email: aishwaryan-ece@dayanandasagar.edu

***Abstract*— Sensitive information, such as financial data, military intelligence, and medical records, is communicated using images. The need for safe digital image encryption techniques has grown as digital photographs are used and transmitted over the internet more and more. In this context, the use of Substitution Box (S-box) combined with dynamic step has proven to be a dependable scheme for image encryption. In S-Box based scheme, a substitution method is employed to replace an image's original pixel values with new values generated from a predefined table. A key-sensitive variable step and bit swapping techniques further encrypt the substituted pixel values. This work examines the algorithm's efficacy using metrics like Entropy and the Number of Pixel Change Rate (NPCR).**

***Index Terms*— S-Box, dynamic online encryption, nibble swapping, pixel substitution, NPCR, Image encryption, Entropy.**

I. INTRODUCTION

According to a report by International Data Corporation (IDC), the world generated 2.5 quintillion bytes of data every day in 2020. This is equivalent to 90% of all the data that has ever been created. The report also predicts that the amount of data generated every day will reach 175 zettabytes by 2025. This is equivalent to 20 quintillion photos or 300 million years of HD video. Digital images are widely used in numerous applications, such as multimedia communication, clinical imaging and security systems. However, the transmission and storage of digital images over the internet are vulnerable to unauthorized access and modification. Therefore, the need for secure digital image encryption techniques has become more essential to guarantee the integrity and confidentiality of digital images.

The main objective of this work is developing a scheme for image encryption using S-Box, key-sensitive online encryption step and nibble swapping techniques to improve the security of digital grayscale images making it challenging for unwarranted parties to access and tweak the images, followed by evaluation of the performance of the algorithm developed.

Employment of S-boxes in block cipher-based encryption is a well-researched for quite some time. Wang ji Jun *et al.*[1] has proposed a multiple simple row-column transform S-box based process with dynamic encryption step to address correlation issues and pixel diffusion for statistical characteristics alteration. In Ref. [2], authors have proposed a S-box using modular design approach, utilizing transformation, modular inverse, and permutation. They also evaluate S-box cryptographic strength. S-box generation for image encryption is proposed in [3]which involves generating a Linear Fractional Transformation (LFT) from a given phrase using the ideas of graph theory. In Ref. [4], rotation matrix that relies on plain image according to different direction is

Grenze ID: 01.GIJET.10.2.608

used for encryption and further block diffusion is utilized to modify the image's statistical characteristics.
Most of these works use complex S-box generation technique to introduce non-linearity. The algorithm reported in this paper attempts to work with simple S-box generation technique and use of nibble swap technique to introduce non-correlation. The paper is further structured in the following manner. Section II covers the scheme proposed in detail. Experimentation results are discussed in Section III. Section IV concludes.

II. Proposed Method

The generic block diagram and the flowchart are as illustrated in Fig. 1 and Fig. 2 respectively. The proposed algorithm has four steps: Read the grayscale image and user input 'key' for the encryption, defining S-box, step based on the key, determine the substituting value and point-wise substitution to perform encryption. Evaluate the performance using metrics Entropy, NPCR.

Figure 1. Generic block diagram of the encryption model

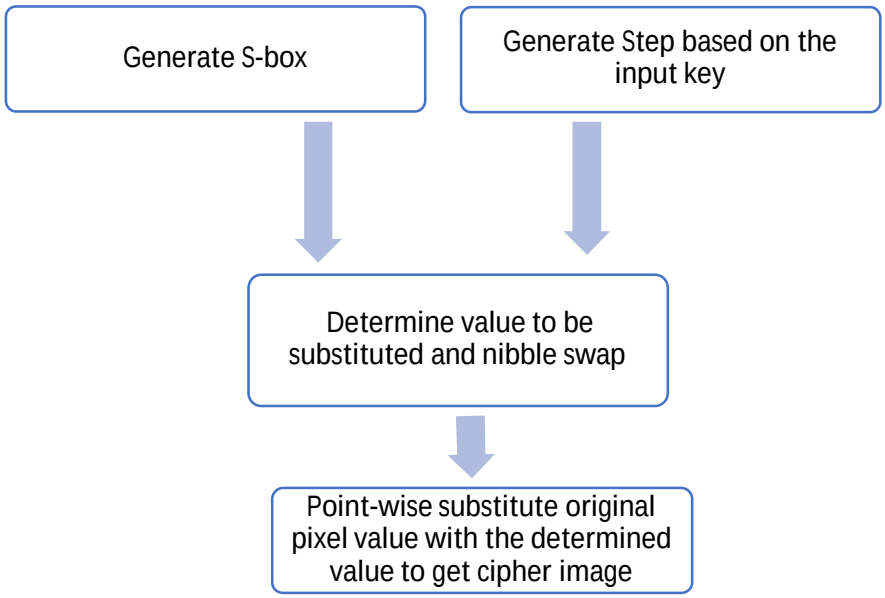


Figure 2. Flow Chart of the Encryption algorithm proposed

A. Reading The Inputs

For any encryption algorithm, an input plain text and at least one key is required. With regards to the proposed image enciphering algorithm, the plain text is a grayscale image. A grayscale image uses a single 2D matrix of size M x N to store pixel values. It is a monochrome image with colours ranging from black (smallest pixel value), shades of gray and white (highest intensity value).
In this work, unsigned eight-bit monochrome images of size 256 x 256, 512 x 512 are used. Since standard images like Lena, Barbara, Cameraman are used, no preprocessing is employed.
The key is utilised to encrypt the plain original text and obtain the cipher text. The key value is variable and its length determines how difficult it is to decrypt. In this work is, the key length is fixed to 10 and is alphanumeric with no special characters and only lower case. Also, for experimentation, a fixed ***key*** is used to encipher.
The size of input and output image will be same.

B. S-Box Generation

The method used to generate is an important aspect in determining the strength of encryption algorithm. This is because S-box introduces non-linearity between the input plain text and output cipher text making it difficult to attack.

In this work, a simple approach is adopted to generate the S-box ***S*** by generating a permutation of integers in the range [0,255] randomly ensuring consistent randomness across different executions if the same ***key*** is provided.

- Size of ***S*** is 1 x 256
- All elements in ***S*** are integers in [0, 255] range
- None of the elements in ***S*** are repeated

C. Defining Step in Real-Time

A dynamic real-time variable ***step*** is generated based on the key value. This further makes reverse engineering difficult. Steps in defining ***step*** are:

$$val = Convert\ \textbf{key}\ into\ an\ integer$$

$$\textbf{step} = val\ \%\ 255\ (where\ \%\ denotes\ modulus) \qquad (1)$$

The ***key*** determines the ***step***. Consequently, the name real-time or dynamic since key is a client input. Additionally, whenever the ***key*** changes, the ***step*** value changes. ***step*** is updated for each pixel value.

D. Performing Encryption

The original pixel value is altered in this step.

Let f be the input image (plain text). $f(x, y)$ indicates the pixel value at (x, y). g be the output image. A generic equation representing the encryption is expressed as

$$g(x, y) = E\{f(x, y)\} \qquad (2)$$

where, E is an iterative encryption function described by the below steps:

i) Obtain the pixel value $p = f(x, y)$

ii) Determine the substitution value from the S – box

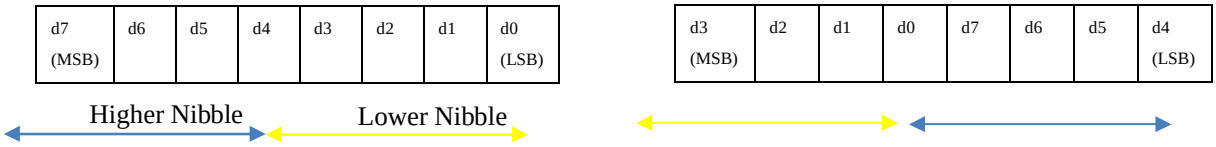
$$index = (p + step)\ \%\ 256$$

$$Sval = S[\ index\ ] \qquad (3)$$

$index$ must be within the limits of size of ***S***

Convert Sval to binary (8- bit) and swap lower and upper nibbles to get $Sval_{NS}$

Sval (Before Nibble swap) Sval $_{NS}$ (After Nibble swap)



iii) Encrypted value is obtained by bit-wise XORing ($\oplus$) the pixel value p and the substitution value determined $Sval_{NS}$

$$g(x, y) = p \oplus Sval_{NS} \qquad (4)$$

This keeps the assailant from getting important data by looking at plain and cipher images.

iv) Update the step for next pixel in the image

$$step = (\ step_{(current)} + Sval\ )\ \%\ 256 \qquad (5)$$

v) Repeat the above steps for all pixel values in f i.e., $\forall x \in [\ 0, M-1]$, $\forall y \in [\ 0, N-1]$.

This results in a scrambled image.

E. Evaluation

The effectiveness of the algorithm is demonstrated using metrics like Entropy, NCR, SSIM and PSNR.

- Entropy: In image encryption, entropy is utilized to estimate the randomness of the encrypted image. The higher the entropy, the more random the image, and the more difficult it is to decrypt. Entropy H is calculated by considering the probability distribution of the pixel values in the image.

$$H(x) = \sum_{i \in x} p(i) \log_2 \frac{1}{p(i)} \quad (6)$$

x represents a set of elements, p(i) indicates the probability associated with each element i in set x.

- Number of Pixel Change Rate (NPCR): It is a metric used to measure the strength of an image encryption algorithm. NPCR is calculated by comparing the number of pixels that have changed betwixt the original and the encrypted images.

$$NPCR = \frac{Number\ of\ changed\ pixels}{Total\ number\ of\ pixels} * 100 \quad (7)$$

- Structural Similarity Index is abbreviated as SSIM. It is a criterion used to measure the similarity between 2 images. In image encryption, SSIM can be utilized to assess the nature of the cipher image.

$$SSIM(x, y) = \frac{(2 * \mu f * \mu g + C1) * (2 * \sigma fg + C2)}{(\mu f^2 + \mu g^2 + C1) * (\sigma f^2 + \sigma g^2 + C2)} \quad (8)$$

where f and g represent the two images being compared (original and encrypted image), μf and μg are the averages of the pixel values in images f and g, respectively. σf^2 and σg^2 are the variances of the pixel values in images f and g, respectively, σfg is the covariance of the pixel values between images f and g, C1 and C2 are constants used to stabilize the division and avoid division by zero errors.

- Peak Signal to Noise Ratio (PSNR) is another metric to measure the quality of the encrypted image. A lower value of PSNR indicates a poor-quality image i.e., the encryption algorithm has introduced sufficient scrambling. PSNR is given by

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \ dB \quad (9)$$

The Mean Square Error (MSE) is given by

$$MSE = \frac{1}{MN} \sum_{i=o}^{M-1} \sum_{j=0}^{N-1} [f(i,j) - g(i,j)]^2 \quad (10)$$

where f represents the matrix pixel data of original image, g represents the matrix data of encrypted image, M and N indicates the numbers of rows and columns in the images respectively, i and j indicates the index of row and column respectively, 255 is the maximum signal value that exists in the original plain image (8-bit.)

III. EXPERIMENT AND RESULTS

The proposed scheme has been simulated in Python based Integrated Development Environment called PyCharm Community Edition 2022.3.2 and tested with unsigned eight-bit grayscale standard images of sizes 256 x 256, 512 x 512.

A. Encryption

The sample output of the proposed encryption algorithm is as illustrated in Fig. 3. The first row consists of the original images and the corresponding encrypted images are in second row. The images used in first row are Lena image (256 x 256), Fruits (256 x 256), Barbara (256 x 256). Row 3 corresponds to Cameraman image (512 x 512) and its encrypted image. The encrypted image appears scrambled, making it challenging to visually discern the underlying data.

B. Entropy

As seen from Table I and Fig. 4, entropy of original 8-bit images is less than 8, however after encrypting it is close to 8 indicating that each pixel value is equally likely which intern implies images are difficult to attack.

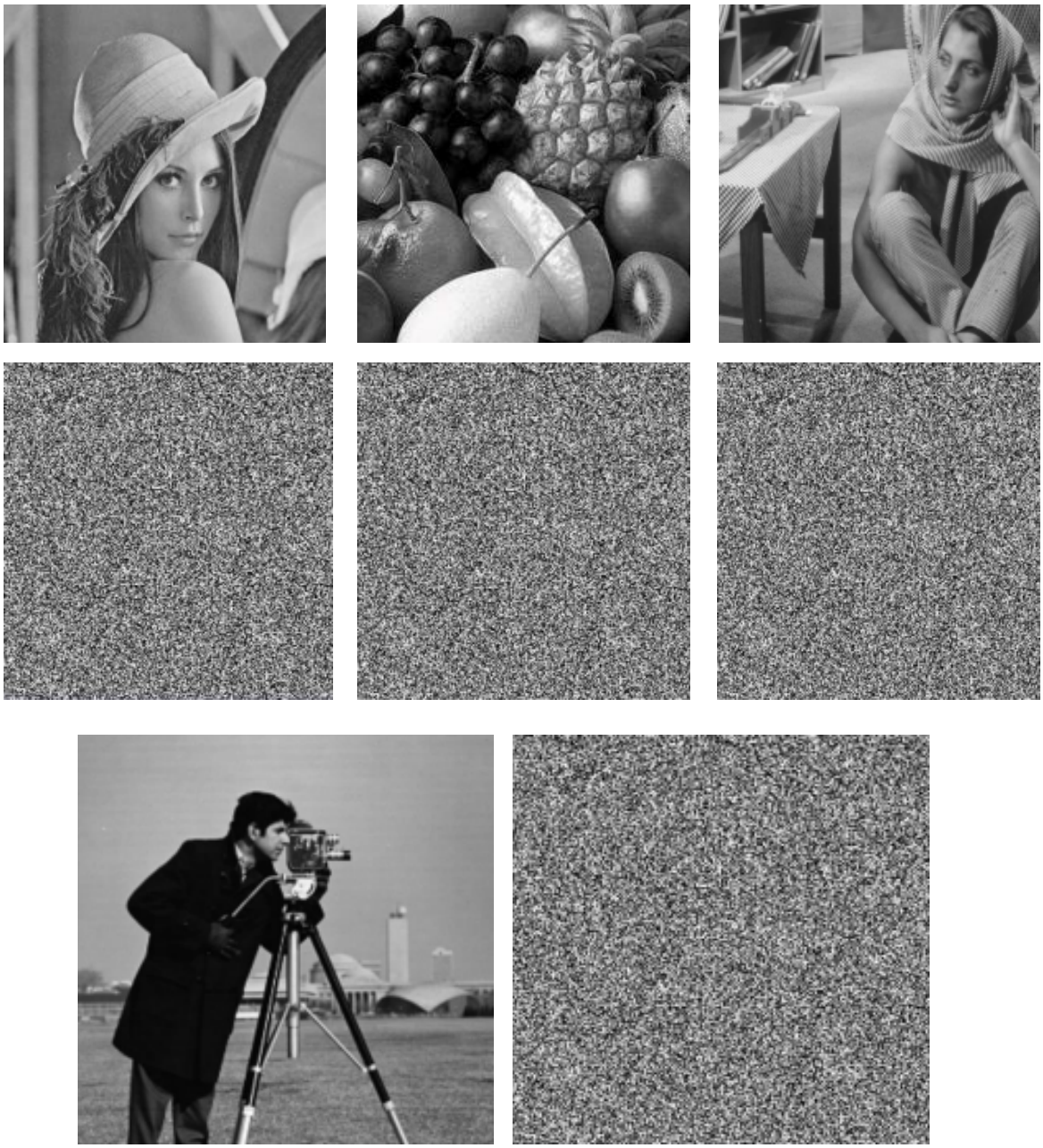


Figure 3. Original Image and Encrypted images

TABLE I. ENTROPY OF IMAGE – BEFORE AD AFTER ENCRYPTION

Image	Entropy (Before Encryption)	Entropy (After Encryption)
Lena	7.442867238	7.997334453
Fruits	7.890455999	7.997078773
Barbara	7.632119011	7.999201021
Bird	6.774394274	7.99557895
Einstein	6.89384322	7.99726949
Cameraman	7.04795523	7.99869056

Figure 4. Plot of Entropy values

TABLE II: EXPERIMENTAL RESULTS - SSIM, NPCR AND PSNR VALUES

Image	SSIM	NPCR	PSNR
Lena	0.0106	99.65668	9.243514
Fruits	0.006021	99.59106	7.991688
Barbara	0.008512	99.62349	8.815332
Bird	0.009612	99.58954	9.484704
Einstein	0.011143	99.51172	9.55518
Camera man	0.00789	99.63531	8.46008
Average values	0.008963	99.6013	8.925083

C.SSIM

From Table II, the SSIM value is close to 0 which indicates the encrypted algorithm as introduced a lot of distortion making it difficult for the hacker to decipher.

D. NPCR

The NPCR values in Table II are close to 100.This indicates that the algorithm provides a high resistance to attacks.

E. PSNR

PSNR values are close less than 10 which indicates that encrypted image and the original image are not similar (Table II.).

IV. CONCLUSIONS

This works paper presents an image encryption algorithm to protect the data in the digital images and hence provide security. The real-time step and nibble-swap techniques further introduces non-linearity. Additionally, step value is a variable which depends on the real-time input key. This makes it difficult to hack. The experimental results on an average are SSIM 0.008963, NPCR and PSNR = 8.925083 which are quite close to the ideal values in the realm of image encryption. Further research can be carried on key space, encryption time along with robustness analysis, real time encryption and decryption on hardware.

REFERENCES

[1] W. J. Jun and T. S. Fun, “A New Image Encryption Algorithm Based on Single S-Box and Dynamic Encryption Step,” IEEE Access, vol. 9, pp. 120596-120612, 2021, doi: 10.1109/ACCESS.2021.3108789.

[2] Amjad Hussain Zahid, Eesa Al-solami, And Musheer Ahmad, “A Novel Modular Approach Based Substitution-Box Design for Image Encryption,” vol.8, July 29 2020, Doi: 10.1109/ACCESS.2020.3016401

[3] Mahboob, A., Nadeem, M. & Rasheed, M.W., “A study of text-theoretical approach to S-box construction with image encryption applications,” Sci Rep 13, 21081 (2023). https://doi.org/10.1038/s41598-023-47607-6
[4] Yushu Zhang, Di Xiao, “An image encryption scheme based on rotation matrix bit-level permutation and block diffusion,” Communications in Nonlinear Science and Numerical Simulation, Volume 19, Issue 1, 2014, pages 74-82, ISSN 1007-5704, https://doi.org/10.1016/j.cnsns.2013.06.031
[5] H. Liu, A. Kadir, and X. Sun, “Chaos-based fast colour image encryption scheme with true random number keys from environmental noise,” IET Image Process., vol. 11, no. 5, pp. 324–332, Jan. 2017.
[6] X.-Y. Wang, S.-X. Gu, and Y.-Q. Zhang, “Novel image encryption algorithm based on cycle shift and chaotic system,” Opt. Lasers Eng., vol. 68, pp. 126–134, May 2015.
[7] Rafael C. Gonzalez and Richard E. Woods. 2006. “Digital Image Processing” (3rd Edition). Prentice-Hall, Inc., USA.
[8] S Jayaraman, S. Esakkirajan, T., Veerakumar, “Digital Image Processing,” 1st edn. McGraw Hill Education 2009.