International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 09 | Sep 2023 www.irjet.net p-ISSN: 2395-0072

Secure Grayscale Image Encryption Using Symmetric Block Cipher with Enhanced Avalanche Effect

Chirag Date[1], Chris Correia[2], Yashvi Vaghela[3], Harshal Vaidya[4]

Abstract - *In today's data-driven landscape, securing digital assets is imperative. Digital images constitute a significant portion of the data transferred online, making them vulnerable to data theft and unauthorized access. This study aims to mitigate these risks by introducing an efficient encryption method. This research addresses the critical need for safeguarding digital image transfers. Employing a Symmetric Block Cipher in Electronic Code Book mode, the algorithm divides images into 64-pixel blocks for encryption. The key generation process incorporates prime numbers and character lists, ensuring a robust encryption key. The algorithm's unique strength lies in its combination of additive and multiplicative ciphers, introducing unpredictability and diffusion. The proposed algorithm achieves an exceptional avalanche effect of 98.48%, highlighting its ability to obscure patterns and relationships between the original and encrypted images. Additionally, the algorithm maintains a flat histogram, indicating a uniform distribution of pixel intensities, a desired characteristic for secure image encryption. In conclusion, the algorithm excels in both security and efficiency, making it suitable for various applications requiring the protection of sensitive image data. Its strong avalanche effect and decryption accuracy provide robust security in the face of evolving cyber threats. Future work could extend this algorithm to RGB images.*

Key Words: **Symmetric Block Cipher**, **Grayscale Image Encryption**, **Avalanche Effect, Data Security**

1.INTRODUCTION

In today's modern world, data has become the lifeblood of society and the economy, shaping the way we live, work, communicate, and make decisions. The need for safe data storage, retrieval, and transfer has also increased along with the exponential growth of data. Ensuring the confidentiality, integrity, and availability of data is not only a technical necessity but a strategic imperative to maintain trust, protect privacy, and enable informed decision-making in this interconnected world.

Digital images constitute a substantial portion of the data being transferred online. Because of their intuitiveness and ease, they have become key data transmission material in the network. Particularly in light of the present degradation in network security, information transmission and sharing based on digital photographs regularly confront issues of data theft, deletion, and assault, which have led to major losses for the owners or publishers of digital images. As a result, it is important to prioritize the secure transfer of these digital images to prevent unauthorized access and potential breaches.

The result of compromised image transfer goes beyond mere financial losses. They encompass a broader spectrum of data security consequences, including identity theft, loss of intellectual property, erosion of trust, and violation of privacy rights. The consequences ripple across individual users, businesses, and organizations alike, impacting their credibility and disrupting their operations. Therefore, establishing secure protocols for transferring digital images takes on paramount significance.

One of the most effective ways to secure data, especially when it comes to the transfer of digital images, is by using encryption. Encryption is a process that transforms data into an unreadable format through the use of complex algorithms. This rendering of data into an unintelligible form serves as a virtual lock, ensuring that only those with authorized access can decrypt and comprehend the information.

Encryption safeguards the confidentiality of images by making it exceedingly difficult for attackers to understand the data. This is particularly crucial when images contain personal information, intellectual property, or any other sensitive content. By rendering images into a scrambled format, encryption ensures that even if data falls into the wrong hands, it remains functionally useless without the corresponding decryption key.

In a world where cyber threats continue to evolve and adapt, encryption stands as a foundational pillar of data security. It instills trust in data exchanges, and as technologies advance and cyber threats become more sophisticated, encryption remains a resilient and versatile tool that can adapt to various scenarios and security requirements.

A common method in cryptographic techniques is cipher block chaining. In the context of digital image transfer, cipher block chaining involves breaking down the image into smaller blocks, and then encrypting them in a way that each block's encryption depends on the previous block's encrypted output. As a result, it becomes more

difficult for attackers to identify patterns in the encrypted data due to the increased interdependence between the blocks.

The advantage of cipher block chaining for images is its resistance to patterns. Without chaining, if two identical blocks of data are encrypted using the same key, they would produce the same ciphertext, potentially revealing information about the data. In cipher block chaining, even if the data blocks are identical, the interdependence introduced by the chaining mechanism ensures that they will have different encrypted forms.

2. LITERATURE REVIEW

Panigrahy, S.K., Acharya, B. and Jena, D [1] employed generating self-invertible matrices for the Hill Cipher algorithm , which removed the computational cost involved in finding the inverse of the matrix while decrypting the message. The research covered the Hill Cipher's drawbacks, including its inability to encrypt pictures with large portions of a single color. The proposed method was tested on different grayscale and color images, and the results were presented.

Acharya, B., Panigrahy, S.K., Patra, S.K. and Panda, G. [2] addresses the challenge of key invertibility and presents a method to generate an involutory key matrix, ensuring decryption without matrix inversion. The algorithm demonstrated effectiveness in encrypting grayscale and color images. It overcame the limitations of the original Hill Cipher, especially for images with uniform backgrounds. The research offered valuable insights for image encryption.

Singh, L.D. and Singh, K.M [3] mapped the pixel values of the image to Elliptic curve coordinates using a lookup table or point multiplication operation with a generator. The data stream from the image is encrypted using a pseudo-random key stream created by a cyclic elliptic curve point and chaotic system. To decrypt the image, a mapping table is required. The Elliptic Curve Discrete Logarithm Problem provides high security with the use of comparatively smaller key sizes to other cryptographic techniques.

Jolfaei, A. and Mirghadri [4] presented a combination of pixel shuffling and a modified version of simplified AES. The encryption scheme uses chaotic baker's map for pixel shuffling and S-AES algorithm for encryption. The algorithm was evaluated through various tests and found to have satisfactory security and efficiency. The cipher has a uniform distribution and is resistant to statistical and known plaintext attacks. The proposed scheme also recorded high key sensitivity, making it difficult for an attacker to break the cipher.

Akkasaligar, P.T. and Biradar, S., 2020 [5] applied cryptography in medical image encryption discusses the challenges of providing security and maintaining confidentiality of medical images during transmission. The research used DNA cryptography for medical image encryption. The study suggested a selective medical image cryptosystem employing a dual hyperchaos map and DNA sequences, since traditional encryption techniques are unable to offer the required level of security for medical images. The research explained the digitized medical image encryption schemes and the entropy measurement used to evaluate the quality of the encryption algorithm.

Krikor, L., Baba, S., Arif, T. and Shaaban, Z. [6] used stream cipher and DCT for encryption of data with visual characteristics. The algorithm introduced a selective encryption method that encrypts only small parts of the image bitstream. The two levels of security for digital image encryption: low-level and high-level security encryption where lower-level degrades the visual quality of image, while the latter scrambled the content completely.

A. B. Mohamed, G. Zaibi and A. Kachouri [7] investigated the application of RC5 and RC6 block ciphers for securing digital image transmission in contexts such as military applications, radar systems, and biometrics. They assessed the encryption efficiency of the algorithms, performed a thorough security analysis including key space analysis and differential attacks, and evaluated their performance in the presence of errors in ambient noise. The results demonstrated that while RC6 offers higher security, it comes at the cost of increased energy consumption and packet retransmission in noisy environments. The research emphasizes the need for a balanced approach in choosing encryption algorithms, considering both security and energy efficiency, particularly in applications like wireless sensor networks.

S. Sangewar and S. Gugulothu [8] presented a method for securing image transmission using visual cryptography. It addresses the challenge of transmitting confidential images securely over a network. The method involves splitting two confidential images, overlaying the portions, and then encrypting the resulting image. RSA algorithm and hashing methods are employed in the encryption process. At the receiver end, decryption requires a key provided by the sender.

3. METHODOLOGY

The proposed algorithm is a block cipher in Electronic Code Book mode. The algorithm makes use of three unique lists, each with a particular function. These lists are essential components of the algorithm.

- `prime_list`: This list contains a series of prime numbers. These primes are used as multipliers in the multiplicative cipher, a crucial component of the encryption process.

- `multi_inv_list`: This list holds the multiplicative inverses corresponding to the prime numbers in `prime_list`. Multiplicative inverses are essential for decryption.

- `char_list`: A list of characters, including digits and uppercase letters. It's used to generate a random encryption key.

3.1 Key generation

In key generation, the algorithm initiates by generating a 64-bit random encryption key, utilizing characters from char_list. This key is fundamental to the encryption and decryption procedures, assuring the security of the data. The randomness and length of this key contribute significantly to the robustness of the encryption scheme, making it exceedingly challenging for unauthorized parties to decipher the protected information.

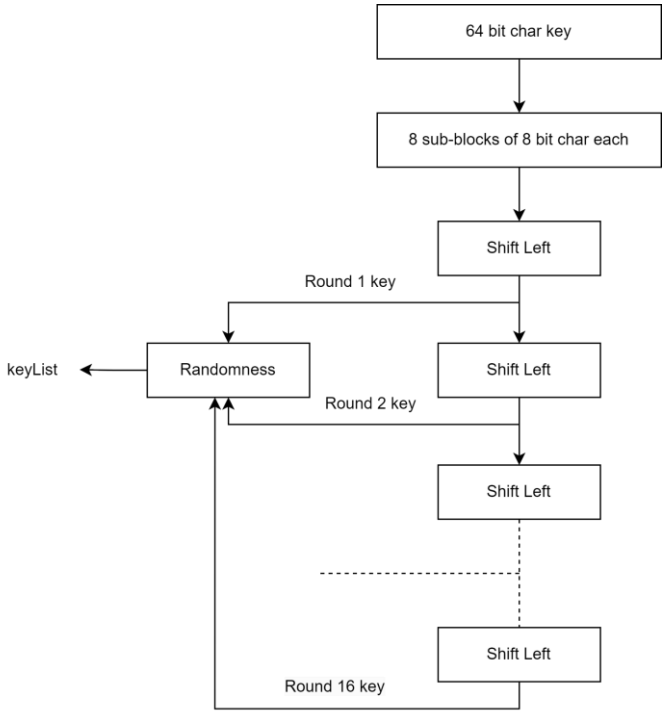


Fig 1. Proposed Key Generation algorithm

The encryption key is subdivided into 16 subkeys. Each subkey is an 8-character string extracted from the original key. The function repeatedly does a circular left shift operation on the 64 bit key which is passed through a randomness function to generate subkeys. The number of times shifting operations are performed depends on the round number as shown in Table 1. The randomness function introduces a degree of unpredictability in the encryption process. a random bit position is selected within the key, encompassing all 64 bits, and is designated for modification. Subsequently, the chosen bit is toggled or "flipped" using the flip_bit function, which essentially inverts its value. This seemingly minor alteration is instrumental in bolstering the algorithm's resistance against attacks and enhancing its avalanche effect.

Key generation shifting	
Rounds	Shift
2,5,9,13	2 char circular left shift
Other	1 char circular left shift

Table -1: Key generation shifting operation

During each round, a random bit of key is replaced by a random value from the 'char_list' using the flip-bit function. Introducing randomness into the key ensures that even if an attacker has some knowledge of the key expansion process, they cannot easily predict the exact subkeys used in each round, thereby bolstering the overall security of the cryptographic system.

3.2 Encryption scheme

The input image is loaded and represented as a grayscale image, with pixel values ranging from 0 to 255. The dimensions of the image are determined by rows and columns.

The image is divided into 64-pixel blocks. It is done by first converting the 2D image matrix into a 1D list by iterating through rows and columns of the image. This is done to allow the processing of image pixel by pixel.

If the total number of pixels is not evenly divisible by 64, it calculates how many additional pixels are needed to create complete 64-pixel blocks and stores this in extra_pixel. It appends the required number of zero pixels to the end of the 1D list to ensure that the image can be divided evenly into 64-pixel blocks. These zero pixels are used to pad the image. It then iterates through the 1D list and extracts consecutive blocks of 64 pixels, storing each block as a sublist in the input_blocks list. Each block is processed separately during encryption and decryption.

The encryption function takes image blocks, which is a list of 64 integers representing 8x8 pixel values of an image block, and a key, which is a 64-character string used as the encryption key. The input block of 64 pixels is further sliced down to sub-blocks of 8 pixels each It initializes an empty list input_blocks to hold the 8 sub-blocks of the input image block. These sub-blocks are 8 pixels each and

are created by slicing the input_value list into 8 equal parts.

The image blocks go through 16 rounds of encryption, where for each round:

The function iterates through the 8 sub-blocks in input_blocks. For each sub-block as in Fig 3. , a custom encryption function is called which works for each pixel value. The result of this function call is a new 8-pixel sub-block, which overwrites the original sub-block in input_blocks.

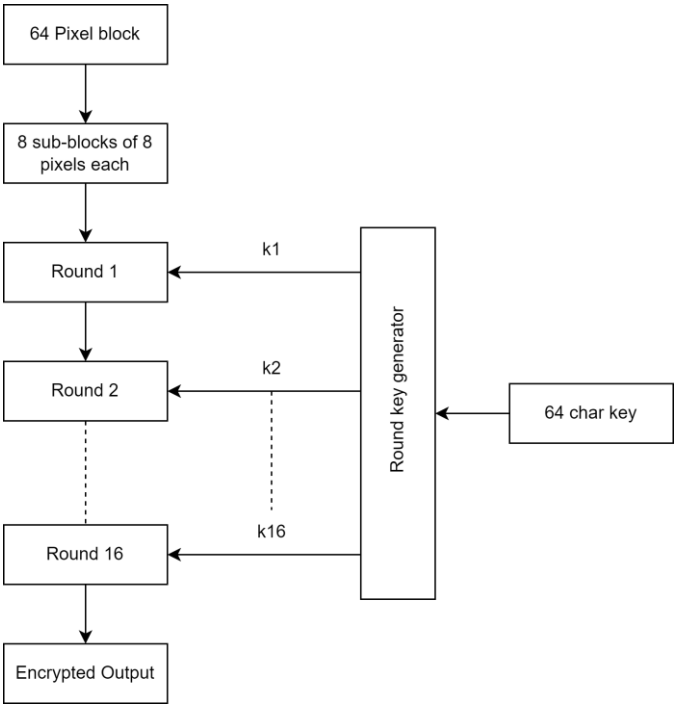


Fig 2. Proposed Encryption Rounds scheme

At the core of the algorithm, there are two pairs of essential functions: OwnEncrypt and OwnDecrypt. These functions are responsible for performing encryption and decryption on 8-character input blocks, employing the provided encryption key.

After each round up of the OwnEncrypt function until the 15th round, it performs a left shift operation on the input_blocks. This left shift operation is a data manipulation step that is common in block ciphers to introduce diffusion.

The custom function combines two types of ciphers, namely additive and multiplicative, to transform the input pixels. The 8-character key is divided into two parts, key1 and key2 containing 4 characters each. Similarly, the 8-character input value is separated into two parts, input1 and input2, with input1 representing the first 4 characters and input2 representing the last 4 characters.

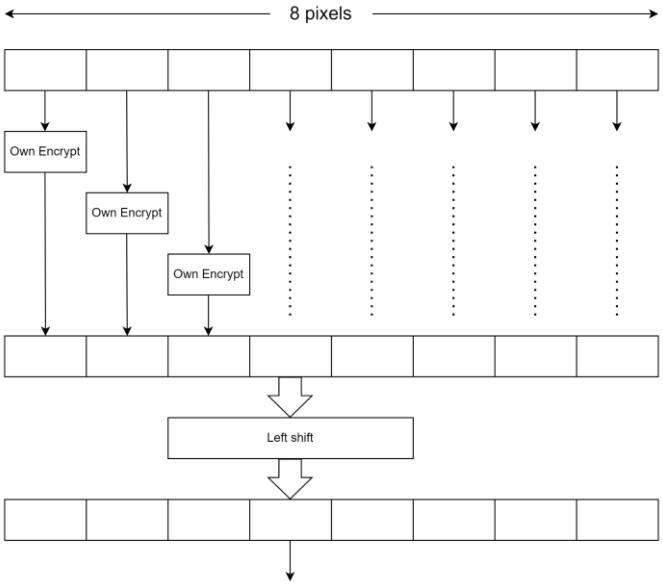


Fig 3. Proposed Pixel Encryption scheme

The first part of input and key goes through an additive cipher in which each character in key1 at the corresponding position is converted into an index value using char_list. The index value obtained is used in the subsequent operation. The character in input is subjected to an addition operation with index value . The result is taken modulo 256, ensuring that the value remains within the byte range.

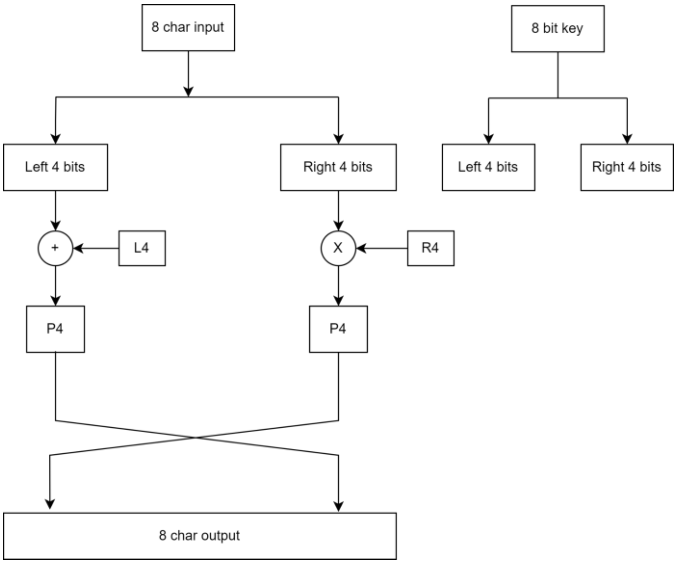


Fig 4 . Proposed OwnEncrypt scheme

The second part of input and key goes through a multiplicative cipher. Similar to the additive cipher, the character in key2 at the corresponding position is converted into an index value using char_list. This index value is used to access a prime number from 'prime_list'. The prime number retrieved from prime_list is used for

multiplicative operation with character of input and the result is taken modulo 256 to ensure the value remains within the byte range. The results of both the additive and multiplicative ciphers go through p4 permutation. P4 Permutation technique reorder bits within a 4-bit input shuffling the bits in a defined manner.

After completing all 16 rounds of encryption, the function flattens the encrypted pixel values into a 64-element list.

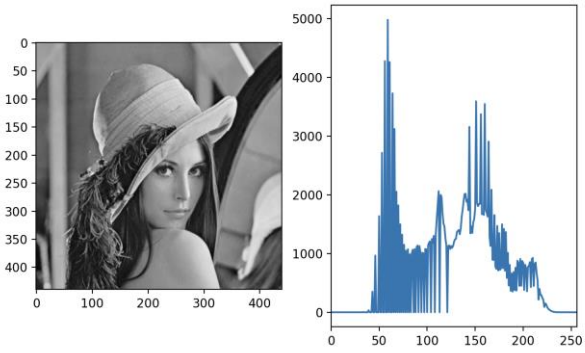
4. RESULT

To assess the algorithm's ability to obscure patterns and relationships between the original grayscale image and its encrypted counterpart, we conducted an avalanche effect analysis. The results are summarized in Table 2.

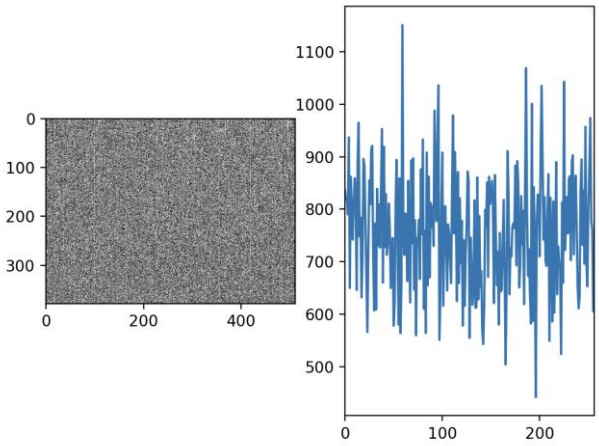
Table -2: Avalanche Effect Percentage Comparison

Average Avalanche Effect Percentage	
J. R. Paragas,A. M. Sison &R. P. Medina(2019) [9]	55.34%
Proposed Algorithm	98.48%

The finding underscores the algorithm's remarkable capability to obscure any discernible patterns or relationships between the original image and its encrypted counterpart. These results underscore the remarkable capability of our algorithm to obscure any discernible patterns or relationships between the original grayscale image and its encrypted counterpart. Even minor adjustments in the input image or encryption key result in substantial modifications in the encrypted image, making it highly resistant to unauthorized decryption attempts.



(A)



(B)

Fig 5. (A) Original Image and its corresponding histogram. (B) Encrypted Image and its corresponding histogram.

Assessing the impact of cryptographic operations on image quality and pixel distribution is paramount. Histogram analysis plays a pivotal role in this evaluation, providing insights into how encryption processes affect the distribution of pixel intensities within a grayscale image. A well-encrypted image should ideally exhibit a uniform distribution with regards to pixel intensities, obscuring any discernible patterns or recognizable features.

Table -3: Correlation Coefficient on Leena image

Leena Image	Horizontal	Vertical	Diagonal
Original Image	**0.9698**	**0.9843**	**0.9551**
Encrypted Image	**0.1366**	**0.3789**	**0.0490**

A correlation value that is close to zero indicates a weak to nonexistent linear relationship. This is normally what you want to achieve when encrypting images. It shows that the pixel values have been altered during the encryption process in a way that makes them seem unrelated to the original image, which is a desired characteristic for secure image encryption.

The exceptional decryption accuracy and strong avalanche effect observed in our algorithm position it as a viable solution for secure image encryption and decryption. Our algorithm's 100% decryption accuracy assures users that their data will remain intact during transmission and storage, reducing the risk of information loss or

corruption. The significant avalanche effect indicates the algorithm's resilience to attacks that attempt to manipulate input data or keys, adding an extra layer of security. This feature enhances the algorithm's suitability for scenarios where adversaries may attempt to breach the encryption.

In conclusion, our algorithm excels in terms of both security and efficiency, making it a strong candidate for safeguarding sensitive image data across various applications.

5. CONCLUSIONS

In conclusion, our proposed algorithm excels in terms of both security and efficiency, making it a strong candidate for safeguarding sensitive image data across various applications. The histogram analysis of the encrypted image further supports the effectiveness of our encryption algorithm. A flat histogram that is in stark contrast to the fluctuating histogram of the original image confirms that our encryption algorithm produces cryptographic images with a very uniform distribution of pixel values.

The algorithm's remarkable capability to obscure patterns, uniform pixel intensity distribution, and weak correlation with the original image underline its effectiveness in secure image encryption.

As data security continues to be of paramount importance in the modern digital landscape, our algorithm provides a reliable means to protect the confidentiality and integrity of grayscale images. Its strong avalanche effect and decryption accuracy reassure users that their data will remain secure, even in the face of evolving cyber threats.

The proposed algorithm can further be extended to RGB images.

REFERENCES

[1] Panigrahy, S.K., Acharya, B. and Jena, D., 2008. Image encryption using self-invertible key matrix of hill cipher algorithm.

[2] Acharya, B., Panigrahy, S.K., Patra, S.K. and Panda, G., 2009. Image encryption using advanced hill cipher algorithm. International Journal of Recent Trends in Engineering, 1(1), pp.663-667.

[3] Singh, L.D. and Singh, K.M., 2015. Image encryption using elliptic curve cryptography. Procedia Computer Science, 54, pp.472-481.

[4] Jolfaei, A. and Mirghadri, A., 2011. Image encryption using chaos and block cipher. Computer and Information Science, 4(1), pp.172-185.

[5] Akkasaligar, P.T. and Biradar, S., 2020. Selective medical image encryption using DNA cryptography. Information Security Journal: A Global Perspective, 29(2), pp.91-101.

[6] Krikor, L., Baba, S., Arif, T. and Shaaban, Z., 2009. Image encryption using DCT and stream cipher. European Journal of Scientific Research, 32(1), pp.47-57.

[7] A. B. Mohamed, G. Zaibi and A. Kachouri, "Implementation of RC5 and RC6 block ciphers on digital images," Eighth International Multi-Conference on Systems, Signals & Devices, Sousse, Tunisia, 2011, pp. 1-6, doi: 10.1109/SSD.2011.5767447.

[8] S. Sangewar and S. Gugulothu, "Securing Images using Encryption & Decryption," 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 480-486, doi: 10.1109/ICCES57224.2023.10192684.

[9] J. R. Paragas, A. M. Sison and R. P. Medina, "An Improved Hill Cipher Algorithm using CBC and Hexadecimal S-Box," 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2019, pp. 77-81, doi: 10.1109/ECICE47484.2019.8942717.