SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I7P124

Volume 11 Issue 7, 271-279, July 2024

Original Article

Color Image Encryption Using Lightweight Cryptography and Genetic Algorithm for Secure Internet of Things

Manoja Kumar Nayak[1], Prasanta Kumar Swain[2]

[1,2]*Department of Computer Application, MSCB University, Odisha, India.*

[2]*Corresponding Author : prasantanou@gmail.com*

Received: 11 May 2024 Revised: 14 June 2024 Accepted: 11 July 2024 Published: 26 July 2024

Abstract *- Image encryption based on a lightweight algorithm is an emerging area of research for the Internet of Things (IoT), where the ubiquity of sensors and intelligence of devices are key fields for maintaining communication. The development of an efficient, lightweight, and secure color image encryption technique becomes essential to maintain security in IoT communication. Existing image encryption techniques, such as RSA and AES, are complicated and with a higher number of rounds, making them computationally expensive and requiring large amounts of memory. Here, an efficient color image encryption algorithm based on a newly developed lightweight key generation procedure is suggested, which uses two major techniques: the genetic as well as the encryption-decryption algorithm. Here, an image is encrypted by applying (<< or >>) left shift or right shift and ($+_2$) binary modulo 2 function, which performs bitwise. For performing encryption operation, a 64-bit block cipher with a 64-bit key is used along with the (<<) left shift and ($+_2$) function. Similarly, for decryption, we used (>>) right shift and ($+_2$) functions. The proposed lightweight algorithm favors reducing the required number of rounds to bring off the lightweight property. Experimental results clearly demonstrate that our procedure is effective by considering correlation, entropy, and image histogram.*

Keywords *- Encryption, IoT, Genetic Algorithm, Decryption, Key.*

1. Introduction

IoT refers to the ability to exchange vast quantities of sensitive digital data (images, audio, video, and text) between smart devices and the internet. According to Forbes, by the end of 2025, 152000 IoT devices will be connected to the internet each minute. The physical and digital world is connected through the use of billions of inter networking sensors.

In the real world, such sensors are installed in equipment and machines. They collect a variety of data, which includes environmental, geographical, medical, industrial, academic, government, daily life data and logistic data [1].

For example, in IoT applications such as smart city or Radio Frequency Identifications (RFID) systems, data is typically acquired from numerous sources held by different domains (e.g. sensors, RFID tags, smartphones and public or private transit providers). Most of the time, crucial data acquired from sensors is changed on a wireless route that is open to everyone with minimal security and privacy. IoT connectivity is prone to eavesdropping in a 5G heterogeneous environment [2].

The protection of users' data and privacy is a major concern in IoT communication. As a result, security and privacy concerns arise throughout communication. Various cryptography mechanisms are designed to protect valuable information from intrusive users or against unauthorized copying and modification. The term "cryptography" refers to "hidden writing". In electronics communication media, everyone wishes to encrypt information so that strictly the intended receiver can decrypt it [3].

A new area of cryptography is lightweight cryptography, which attempts to secure IoT devices with limited resources. The lightweight cryptography models are designed for use in constrained domains such as sensors, smart cards, RFID tags, healthcare devices, etc. On the other hand, the goal of lightweight encryption is to implement it in gadgets with low requirements (storage, processing power and small size of the device or the device's energy availability) [4].

Devices should be validated to ensure the data source in IoT applications due to the sensitive property of the data. As IoT devices lack high computation ability, and have small size, limited memory, less spectrum and limited battery life,

hence these are treated as less resource devices [5]. Furthermore, lightweight cipher must deal with security, cost, and performance limitations [6].

For color images, lightweight encryption techniques are implemented. Color images are created by combining Red, Green, and Blue, the three colors in their lightest and darkest intensities. A color image is created by combining these color intensities. In cryptography, One of the important components that decide the security level is key generation.

In lightweight cryptography there is a need to optimize the number of rounds needed in key generation so that it satisfies the lightweight property. Most of the algorithms use non-optimized methods in key generation in lightweight methods for IoT, which is a research gap in this context.

In this paper, we proposed a color image encryption technique using lightweight cryptography and a genetic algorithm for a secure Internet of Things, which produces a key applying a genetic algorithm by optimizing a number of rounds. To find the optimal solutions to different problems, a familiar heuristic technique is a genetic algorithm. We applied multi-point crossover as well as mutation operations in our proposed work.

The remaining part is presented as Part 2, which depicts a survey of the literature. Part 3 proposed the framework. Part 4 represents the experimental study and the results, and finally, Part 5 gives the conclusion.

2. Survey of Literature

Large sub-keys are produced by the key generation algorithm applying the user's secret key. This algorithm uses sub-keys to allow the bits in the secret key to influence the procedure of encryption in each cycle. In a key generation process for both encrypting the data and decrypting the data only one key is used. Block size, key length, key scheduling, number of rounds, and operations are discovered to be critical parameters of lightweight block ciphers [7].

Because lightweight cryptography algorithms are chosen for devices with limited resources, the implementation of data security should be improved. The main goal of making block cipher in a lightweight mode is to protect confidentiality and integrity. We began an initiative to add block cipher in lightweight mode, which is described in several papers, by evaluating key characteristics such as block size, key size, architecture, and the number of rounds. Rijndael proposes a block cipher which requires a code with reduced size [8].

It is unaffected by linear cryptography and differential attacks. When the number of rounds is less than 9, the key attack is available. It acts as an energy-efficient encryption when the parameters are optimized.

In [9], the authors present the Skipjack algorithm, which uses an 80-bit key in a 64-bit data block cipher. The space required is minimal because it requires the shortest extended key, less memory and code. This encryption requires low power as compared to Rijndael; however when speed is adjusted, the cipher is more efficient.

The fact that the National Security Agency has not certified it secure. Skipjack was best attacked in 1999 with less than 32 rounds, which is known as Biham. The Skipjack is safe with all 32 rounds. RC5 is an encryption process that is written in the form of RC5-w/r/b. Here, w indicates word size, round numbers are denoted by r and the length of the key is denoted by b [10].

RC5 has low energy efficiency because it uses MSP430F149 of the Achilles heel for rotations and multiplication. RC5 requires a tiny amount of code memory. It was attacked in 1998, and only 244 plain texts were chosen to break RC5-32/12/16. For security reasons, in this case, total rounds should be more than / equal to 18. The 6-round Feistel structure cipher Camellia requires 13 by 28 plain text and has a cipher execution complexity of 2112 [11].

It is made up of 128 bits of data and a 128/192/256-bit key. In comparison with Camellia, the RC5 and RC6 are less energy efficient. Additional memory and a larger code size were required for this. Sequential attack is the name given to the attack on the Camellia.

The AVR Atmel is used to run the resource-limited application in lightweight algorithms such as KLEIN, HEIGHT, KATAN as well as TEA. A microcontroller of small size is used to analyze the efficiency of memory, energy utilization, and confusion level as well as diffusion for security inquiries [12].

In [13], the researchers introduced an ultra-lightweight encryption, which is a 31-round SP network called PRESENT. It has a modest key of size 80 bits with a block of size 64 bits to achieve efficient power usage with a small battery. The DES encryption algorithm uses a 16-round key. A 56-bit key is used in the Feistel encryption, which has a block length of 64 bits [14].

Because the key is 56 bits long, it is vulnerable to an extensive attack that is capable of breaking DES. As a result, DES is no longer secure. Three-DES is more secure because of 48 rounds of security features. SIT is a block cipher technique in which an employee's bits of information is 64, and a key is 64 bits [16].

In the AES block cipher, the Substitution-Permutation (SP) link is used to incorporate Shannon’s confusion-diffusion concept. The Blowfish and DES use Feistel structure to avail the benefits of similar encryption and decryption procedures.

SIT is a security system which achieves security using Feistel Structure and SP Network techniques. There are two phases to this procedure: key generation and encryption. The 64-bit usage key is split into four blocks.

Each block acts as an input for the SP network (here, the F-function) structured in 4x4 matrices to produce five new unique keys by repeating encryption times [15] proposes a technique that combines good confusion and diffusion features. Due to its incredibly vast keyspace, this cryptosystem has a better level of security. Because the encryption and decryption techniques are symmetric, it is used to encrypt color images. In light weight cryptography, this approach is not yet employed.

3. Proposed Framework

We have suggested a color image encryption technique for lightweight cryptography with a genetic algorithm for secure IoT to maintain safe Internet of Things communication. This requires a message of the unencrypted text, and it uses a symmetric key block cipher of 64-bit, which is shown in the algorithm proposed. The sequence of the proposed technique is as follows:

1) Genetic Algorithm (GA) function used for Key Generation,
2) Encryption, and
3) Decryption.

The framework of the proposed method is described in Figure 1.

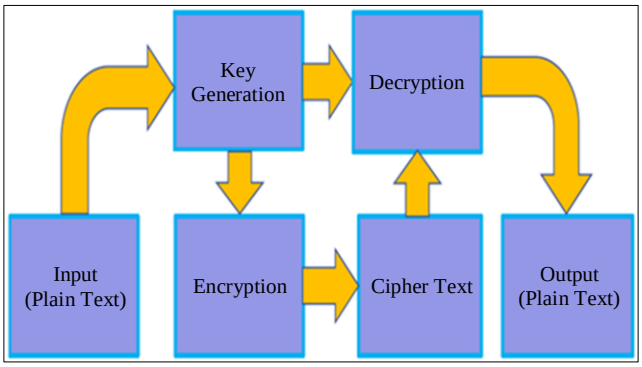

Fig. 1 Overview of the proposed framework

3.1. Key Generation Algorithm

The steps of the key creation block are presented in Figure 2. The key creation part is termed as most important component of the encryption and decryption procedure. Feistel-based encryption method that requires a different key for each round and is dependent on the number of rounds, which is used by several conventional algorithms. On the other hand, the newly proposed lightweight technique required approximately five rounds to both encrypt and decrypt an image.

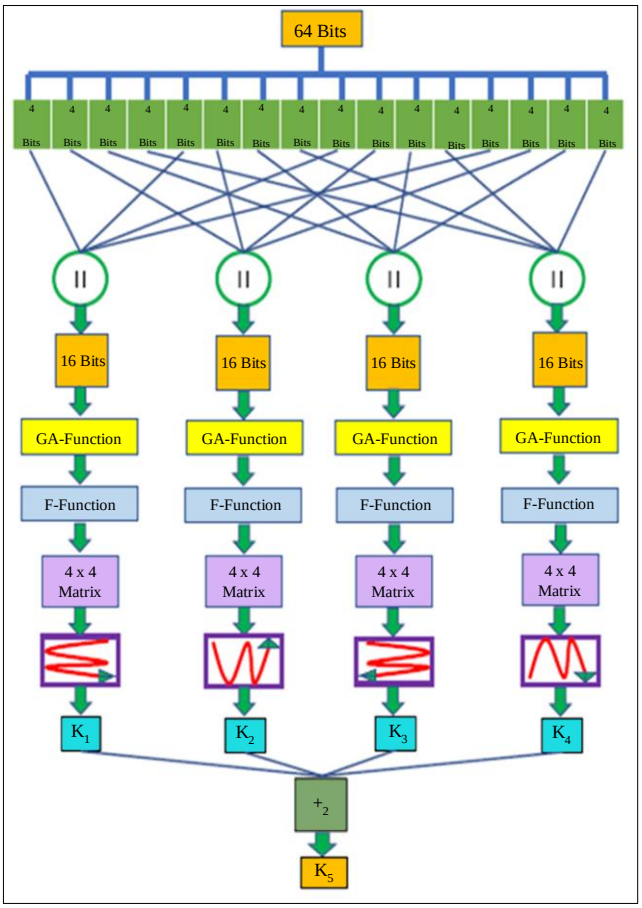

Fig. 2 Key creation method

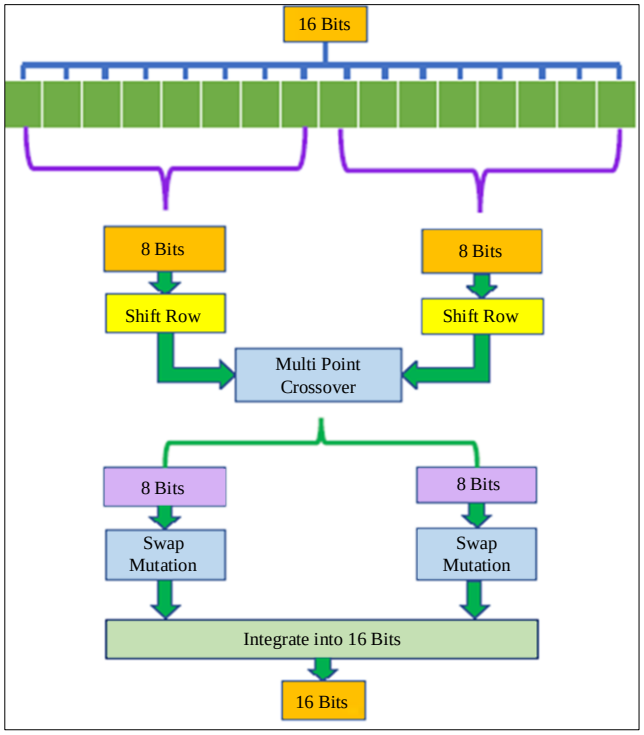

Fig. 3 GA functions

The following explains the key generation process:

- After receiving the 64-bit key (K_c), the key-generating block creates five distinct keys. (as represented in Figure 2).
- The 64-bit K_c is split into four 4-bit segments.
- After making segmentation, two basic operations of GA, crossover and mutation, are used on 4 segments and named GA-function. The GA function performed on 16-bit data is shown in Figure 3.
- The 16-bit input will be split into two 8-bit blocks and supplied into the GA function.
- After the input of 16-bit was split into two equal halves, a shift row is made for both the 8 bits and the multi-point crossover of the GA function accepts the shifting of output.
- The two 8-bit blocks are used to execute the multi-point crossover. The n random combination points are used by this multipoint crossover operator, where n is equal to 4 in our case. The work that comes after is an exchange mutation that exchanges the parameters. In exchange mutation exchange of two random bit happens. Swapping mutation with secure key generation works superior compared to other examined mutation processes. After the exchange mutation process, a 16-bit block is generated by combining two 8-bit blocks. (as shown in Figure 3).
- The F-function (presented in Figure 4 [16]) operates on 16-bit data. First, the substitution of segments of cipher key (K_c) is done, then the four F-function blocks, each of 16-bits, are acquired and defined as:

$$K_{d_i} f = \prod_{j=1}^{4} K_{c(j-1)+i} \quad (1)$$

Here, i vary from 1 to 4 for the initial key generation in 4 rounds (presented in Figure 4).

- Then, the f-function is processed with $K_{d_i} f$, which is 16-bit and produces Kei as shown below:

$$K_{e_i} f = f(K_{d_i} f) \quad (2)$$

- Implementation of transformations such as linear and non linear type using F-Function causes confusion and diffusion. This includes both the P table and Q table shown in Figure 4 [16]. Table 1 has a Khazads mini-box for P, while Table 2 has a Khazads mini-box for Q [17]. The authors in [16] explained the outcome of the total F-function, which is structured into 4 by 4 matrices known as K_m.
- The first 4 rounds are completed by changing matrices into 4 (16-bit) arrays called round keys (K_r) (here it is K_1, K_2, K_3 and K_4). The bits are organized and shown [16]. After that, the $+_2$ function is performed on the round keys (K_1 to K_4) to obtain K_5. This is defined as:

$$K_5 = +_2 \sum_{i=1}^{4} K_i \quad (3)$$

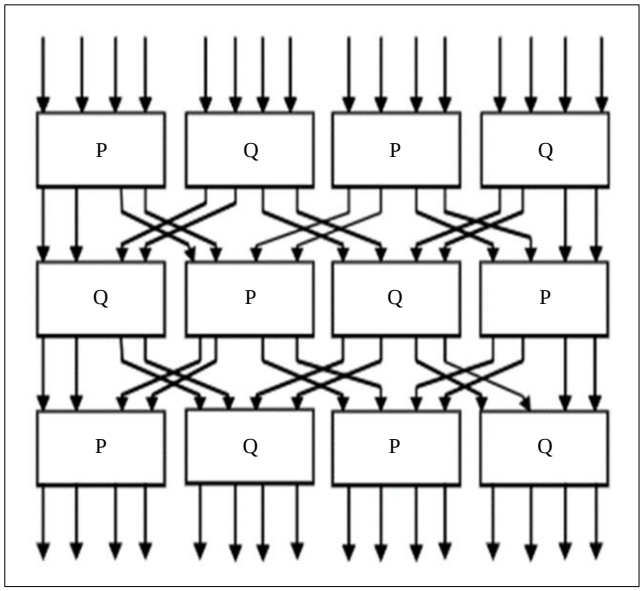

Fig. 4 Functions of the SIT algorithm

Table 1. Correlation analysis of eight plain images

Image	Color Components								
	C.E. Dong [18]			Usman et al. [16]			Proposed (Ours)		
	Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal
Lena	0.9230	0.9271	0.9921	0.9763	0.9698	0.9543	0.9866	0.9768	0.9734
Pepper	0.9501	0.9640	0.9895	0.9860	0.9893	0.9914	0.9891	0.9913	0.9951
Baboon	0.8801	0.9382	0.5351	0.9801	0.9789	0.9770	0.9903	0.9871	0.9831
Airplane	0.8697	0.9174	0.9067	0.9784	0.9758	0.9742	0.9821	0.9885	0.9812
House	0.9414	0.9588	0.9370	0.9541	0.9650	0.9737	0.9634	0.9715	0.9773
Sailboat	0.9219	0.9187	0.9492	0.9573	0.9558	0.9642	0.9633	0.9618	0.9661
Barbara	0.8926	0.8377	0.9846	0.9545	0.9471	0.9560	0.9654	0.9511	0.9603
Boats	0.9239	0.9907	0.9366	0.9588	0.9458	0.9653	0.9619	0.9545	0.9763

Table 2. Entropy analysis of eight plain images (Re-Red, Gr-green, Bl-Blue)

Image	Color Representations								
	C.E. Dong [18]			Usman et al. [16]			Proposed (Ours)		
	Re	Gr	Bl	Re	Gr	Bl	Re	Gr	Bl
Lena	7.9901	7.9912	7.9921	7.9976	7.9974	7.9968	7.9984	7.9993	7.9981
Pepper	7.9893	7.9897	7.9895	7.6876	7.6866	7.7017	7.7813	7.7811	7.7977
Baboon	7.9892	7.9895	7.9899	7.9500	7.9491	7.9456	7.9610	7.9513	7.9561
Airplane	7.9895	7.9894	7.9892	7.4799	7.5154	7.4868	7.5696	7.6541	7.5683
House	7.9902	7.9890	7.9894	7.9155	7.9166	7.9247	7.9359	7.9336	7.9558
Sailboat	7.9890	7.9898	7.9896	7.9958	7.9964	7.9943	7.9991	7.9978	7.9951
Barbara	7.9905	7.9904	7.9906	7.9972	7.9975	7.9965	7.9991	7.9981	7.9974
Boats	7.9908	7.9907	7.9899	7.9374	7.9464	7.9481	7.9541	7.9611	7.9553

3.2. Encryption

The structure that represents the encryption process is represented in Figure 5. Here, the input messages are encrypted into blocks of plain text of size 64 bits in length. The procedures outlined below are executed in order to encrypt the data.

1. To start the encryption process, the key generation process produces several round keys.
2. Initially, the application of a 64-bit plain text array (P_i) is done in the first round as well as split into four 16-bit text segments, indicated as $P_{X0–15}$, $P_{x16–31}$, $P_{x32–47}$, and $P_{x48–63}$.
3. To minimize the identity of the data and increase the confusion of the generated cipher text, each round, the bits undergo the swapping operation. The bit-wise XNOR is performed by round key K_i and generates the key from the process of key generation, denoted by $Px_{0–15}$. The approach for generating Ro_{11} and Ro_{14} results, respectively, is the same for between K_i and $P_{x48–63}$.
4. The F-function receives the outcome of the XNOR function, and the result is produced as Ef_{l1} and Ef_{r1}, which are presented in Figure 5. Between Ef_{l1} and $P_{x32–47}$, the left move function (<<) and binary modulo 2 ($+_2$) function are used to obtain Ro_{12}, and between Ef_{r1} and $P_{x16–31}$, Ro_{13} is obtained. These functions are given by:

$$R_{O_{ij}} = \begin{cases} P_{x_{i,j}} \odot K_i & j = 1 \; and \; 4 \\ \left(P_{x_{i,j+1}} \ll q\right) +_2 Ef_{l_i} & j = 2 \\ \left(P_{x_{i,j-1}} \ll q\right) +_2 Ef_{r_i} & j = 3 \end{cases} \quad (4)$$

5. It used (Equation 4) to replicate the same steps for the remaining rounds. The end round's results are combined to produce the Cipher Text (C_i), which is given by:

$$C_i = R_{51}R_{52}R_{53}R_{54} \quad (5)$$

3.3. Decryption

We have used the symmetric key for performing the decryption of the cipher image so that it can be converted to a plain image. Distinct keys are used that are generated during the key generation process, as in the proposed algorithm. Here diffusion method is used and substitution operations are done. Also, XNOR with (>>) right move operator, ($+_2$) binary modulo 2 function, exchange and left move process are done sequentially.

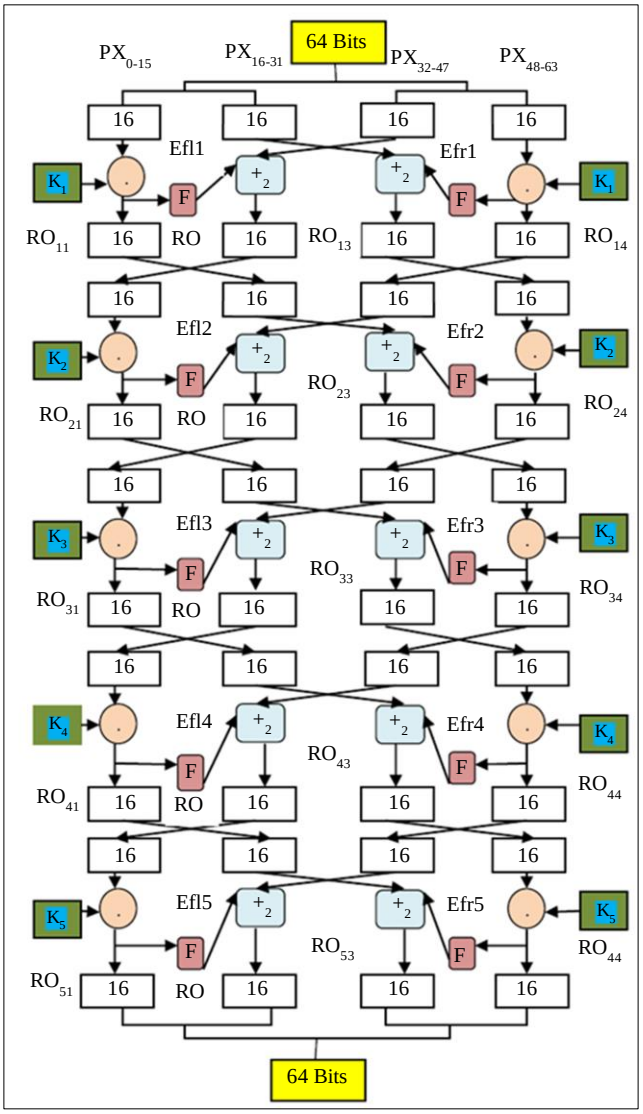

Fig. 5 Flowchart of the encryption process

4. Experimental Study and Results

To prove our suggested model, we selected 8 sample images from SIT to experiment [16, 18]. The above-selected sample is in PNG format; and each one being of size 256×256 pixels. The workability of the cipher outcome is evaluated by correlation, histogram analysis, entropy and differential analysis. A variety of novel methods are used to compare the results [16, 18]. We performed the proposed experiments on a system configured with memory of DDR2 8GB and an Intel-core-i5 processor with a speed of 3.10 GHz.

4.1. Performance Evaluation

4.1.1. Correlation

It is generally accepted that the correlation coefficient is an essential parameter for images that are original and encrypted. The final result indicates that two neighboring pixels show a well-built correlation in the plain image. Hence, the proposed algorithm's influence on encryption is delightful. The correlation coefficient is obtained using the following formula.

$$r_{xy} = \frac{Con(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad (6)$$

Where,

$$\mathrm{Con}(x, y) = (1/N) \sum_{i=1}^{N} \left(x_i - E(x)\right)\left(y_i - E(y)\right),$$

$$E(x) = (1/N) \sum_{i=1}^{N} (x_i), \text{ and}$$

$$D(x) = (1/N) \sum_{i=1}^{N} (x_i - E(x))^2$$

Here, N can be treated as the gross amount of pixels that the image contains, N = row × col, x is a vector of length N, and x_i is the original image with i^{th} intensity values. In this case, the real image has a higher correlation coefficient and was found to be strongly correlated (represented in Table 1). However, there appears to be no correlation with the encrypted image.

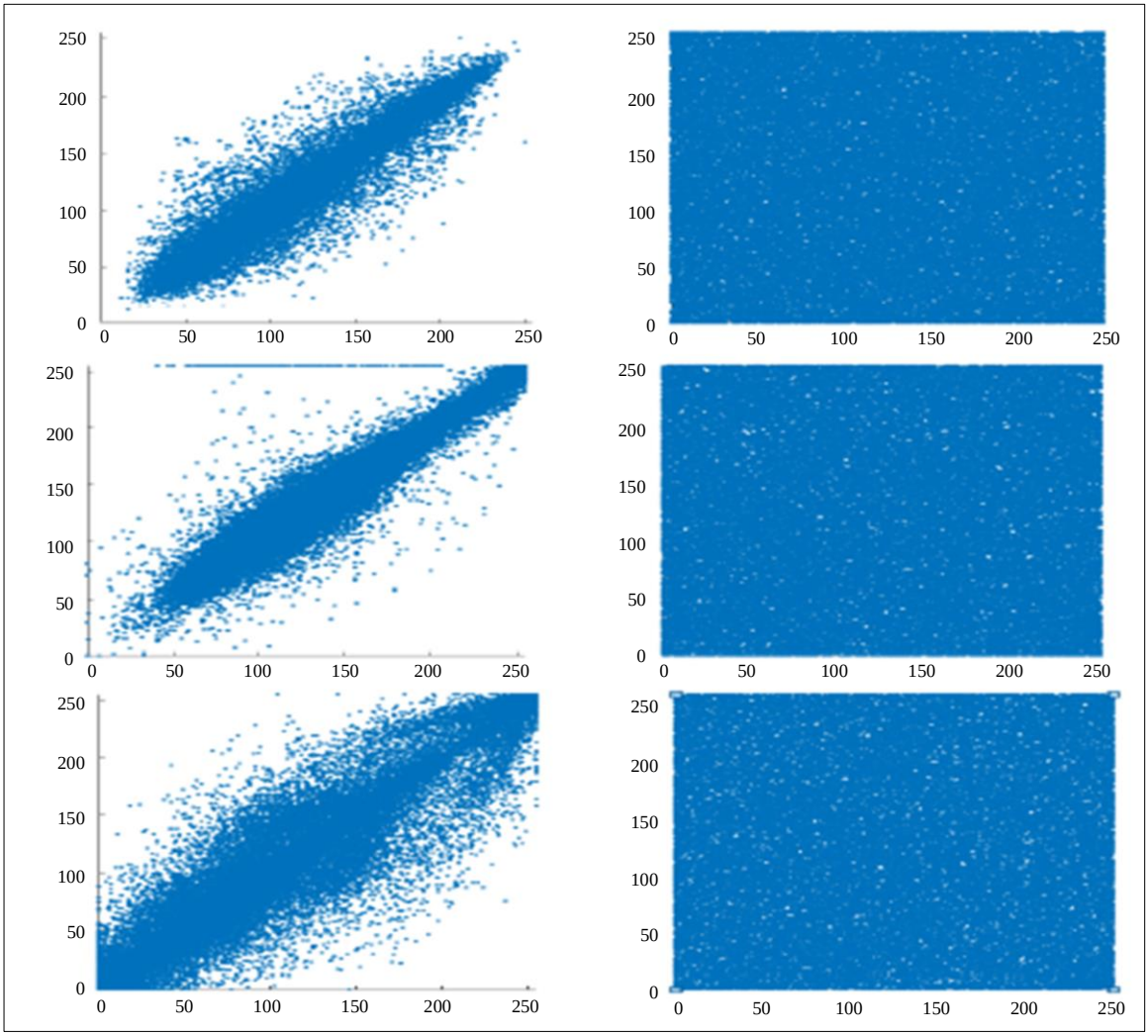

Fig. 6 Correlations for encrypted and decrypted image, (a) Lena image, (b) Baboon, and (c) House.

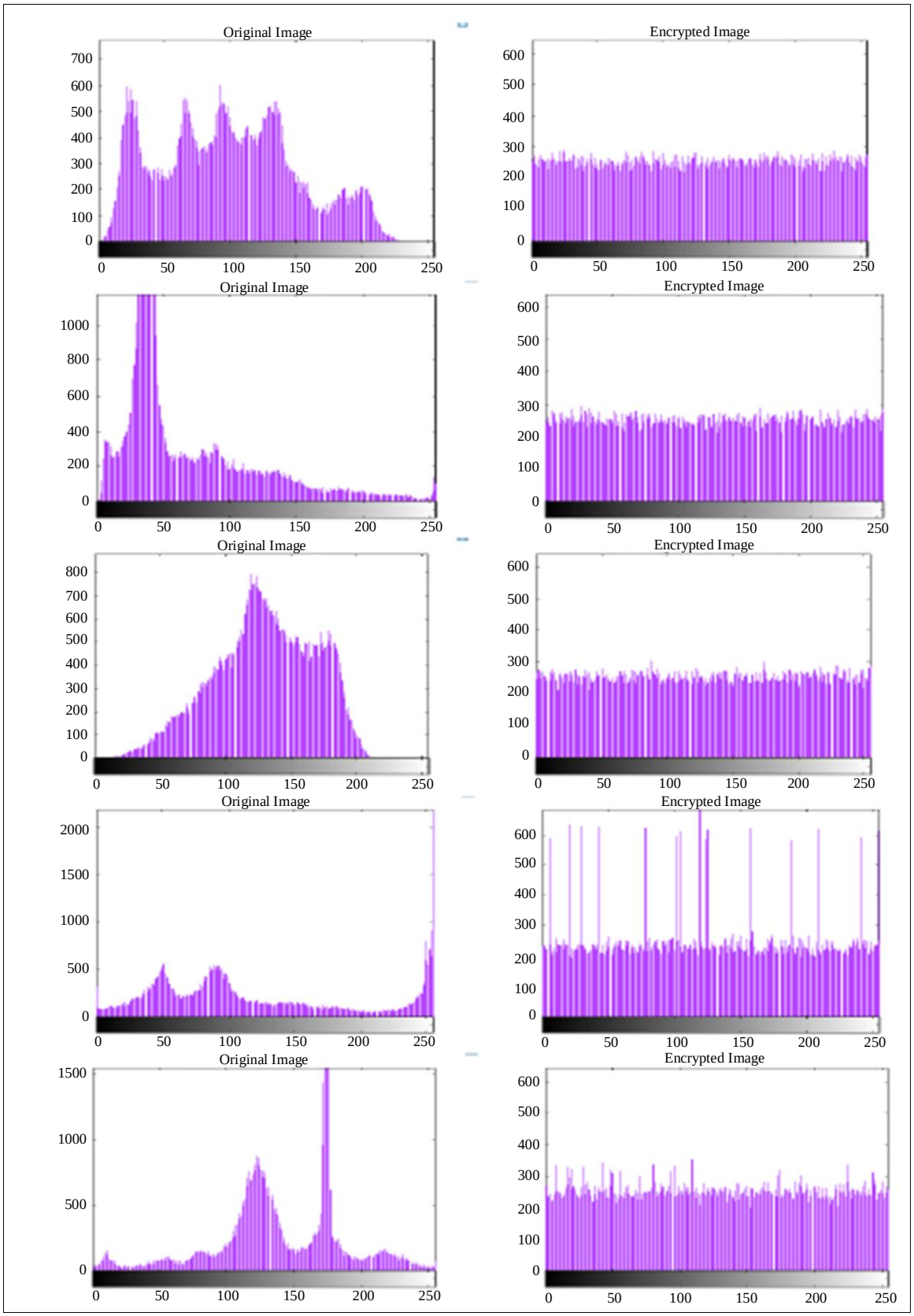

Fig. 7 Image histogram, (a) Lena, (b) Pepper, (c) Baboon, (d) House, and (e) Boat.
(The original image is shown in column-1and column-2 shows x –axis intensity and y-axis frequency)

Table 3. NCPR results of eight encrypted images (Re-Red, Gr-green, Bl-Blue)

Image	Color Components								
	C.E. Dong [18]			Usman et al. [16]			Proposed (Ours)		
	Re	Gr	Bl	Re	Gr	Bl	Re	Gr	Bl
Lena	99.6013	99.6131	99.6226	99.6201	99.6216	99.5911	99.6511	99.6521	99.6901
Pepper	99.6015	99.6282	99.6187	99.7009	99.6582	99.6780	99.7113	99.7122	99.7110
Baboon	99.6053	99.6125	99.6162	99.6246	99.5941	99.6078	99.6813	99.6199	99.6912
Airplane	99.5978	99.6074	99.6074	99.7025	99.6628	99.6964	99.7712	99.6913	99.7223
House	99.6119	99.6074	99.6240	99.6414	99.5956	99.6017	99.6841	99.6009	99.6219
Sailboat	99.6387	99.5952	99.6129	99.6292	99.6078	99.6124	99.6723	99.6308	99.6779
Barbara	99.6278	99.6145	99.6096	99.6277	99.6033	99.6216	99.6771	99.6418	99.6776
Boats	99.6257	99.6193	99.6128	99.6109	99.6201	99.6201	99.6119	99.6300	99.6304

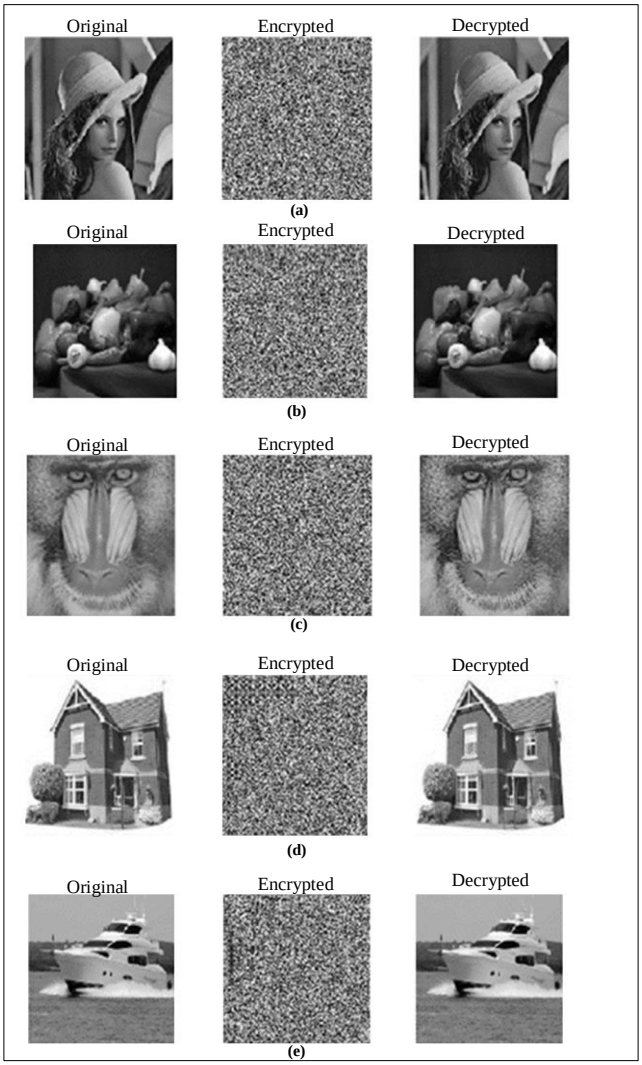

Fig. 8(a) Lena, (b) Pepper, (c) Baboon, (4) House, and (5) Boat-encryption and decryption images.

4.1.2. Image Entropy

The information entropy of randomness is its most basic property**.** Assuming m as the information source, the equation that follows may be used to calculate information entropy:

$$E(m) = \sum_{i=0}^{2^n-1} p(m_i)\log_2 \frac{1}{p(m_i)} \quad (7)$$

Here probability factor for m is expressed by $p(m_i)$. If each data source's 28 states has an equal probability of occurring, we can apply Equation 7 to obtain the ideal E(m) = 8, which indicates randomness of data. This produces the entropy information of the encrypted image roughly at 8 after encryption. As it approaches closer to 8, the probability is that the cryptographic system will show a drop in data. Equation 7 gives the entropy information of the encrypted images. Table 2 displays the entropy values for each of the segments. The results indicate that the entropy of every encrypted image is nearly at maximum (about 8). The higher the entropy, the better the security algorithm works.

4.1.3. Image Histogram

A histogram that displays predictable patterns (flat encrypted image histograms), the high security is indicated by the following encryption. (security breaches are impossible). Figure 7 shows the histograms for the original picture source Lena and Cameraheman to test on the left side. The Histogram of encrypted images is identical on the right side. It is observed that the encrypted image has a uniform distribution of pixels.

4.1.4. Differential Analysis

The set of metrics used to examine the sensitivity of encryption algorithms is the NPCR index and is defined as:

$$NPCR = \frac{\sum_{ij} D(i,j)}{m \times n} \times 100\% \quad (8)$$

$$D(i,j) = \begin{cases} 0 \text{ if } C_1(i,j) = C_2(i,j) \\ 1 \text{ otherwise} \end{cases} \quad (9)$$

Where, C1 and C2 each have a single distinct pixel value to represent the cipher images in relationship with the same plain image.

NPCR is more accurate since it can mathematically demonstrate the sensitivity of plain images. Informally, a high NPCR score is frequently equated to high differential attack

resistance. The NPCR values of the three components are shown in Table 3. The encrypted and decrypted image is shown in Figure 8.

5. Conclusion

The evolution of smartphones and ubiquitous computing like wireless sensor networks, RFID and embedded technologies promotes the IoT process for developing smart technologies. Data security is a serious concern as IoT is employed in many applications such as industrial, healthcare, agriculture, smart cities, etc. To add one step more we proposed a Color Image Encryption technique using Lightweight Cryptography and Genetic Algorithm in IoT security. This method proved to perform better than the existing methods. The entropy generated shows a higher value that indicates strong encryption of images.

References

[1] Daqiang Zhang et al., "NextMe: Localization Using Cellular Traces in Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 302-312, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[2] Deepti Sehrawat, and Nasib Singh Gill, "Lightweight Block Ciphers for IoT Based Applications: A Review," *International Journal of Applied Engineering Research*, vol. 13, no. 5, pp. 2258-2270, 2018. [Google Scholar] [Publisher Link]

[3] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, Delhi, India, 2007. [Google Scholar]

[4] Sezer Toprak et al., "LWE: An Energy-Efficient Lightweight Encryption Algorithm for Medical Sensors and IoT Devices," *Electrica*, vol. 20, no. 1, pp. 71-80, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Sohel Rana et al., "An Effective Lightweight Cryptographic Algorithm to Secure Resource-constrained Devices," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 11, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Seddiq Abd Al-Rahman, Ali Sagheer, and Omar Dawood, "NVLC: New Variant Lightweight Cryptography Algorithm for Internet of Things," *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, Fallujah, Iraq, pp. 176-181, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mickaël Cazorla, Kevin Marquet, and Marine Minier, "Survey and Benchmark of Lightweight Block Ciphers for Wireless Sensor Networks," *2013 International Conference on Security and Cryptography (SECRYPT)*, Reykjavik, Iceland, pp. 1-6, 2013. [Google Scholar] [Publisher Link]

[8] Joan Daemen, and Vincent Rijmen, "AES Proposal: Rijndael," *AES Proposal*, pp. 1-45, 1999. [Google Scholar] [Publisher Link]

[9] Eli Biham, Alex Biryukov, and Adi Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials," *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, vol. 1592, pp. 12-23, 1999. [CrossRef] [Google Scholar] [Publisher Link]

[10] Alex Biryukov, and Eyal Kushilevitz, "Improved Cryptanalysis of RC5," *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, vol. 1403, pp. 85-99, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yeping He, and Sihan Qing, "Square Attack on Reduced Camellia Cipher," *Proceedings of the International Conference on Information and Communications Security*, Springer, Berlin, Heidelberg, vol. 2229, pp. 238-245, 2001. [CrossRef] [Publisher Link]

[12] V.K. Jha, "*Cryptanalysis of Lightweight Block Ciphers*," Master's Thesis, Computer Science and Engineering, Science Degree Programme, Aalto University School and Technology, 2011.

[13] A. Bogdanov et al., "PRESENT: An Ultra-lightweight Block Cipher," *Proceedings of the International Workshop on Crypto Graphic Hardware and Embedded Systems Springer*, Berlin, Heidelberg, vol. 4727, pp. 450-466, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[14] Matt Blaze et al., "*Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security*," Information Assurance Technology Analysis Centre Falls Church VA, Technical Report, 1996. [Google Scholar] [Publisher Link]

[15] Hongjun Liu, and Xingyuan Wang, "Color Image Encryption Based on One-Time Keys and Robust Chaotic Maps," *Computers & Mathematics with Applications*, vol. 59, no. 10, pp. 3320-3327, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[16] Muhammad Usman et al., "SIT: a Lightweight Encryption Algorithm for Secure Internet of Things," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 1, pp. 402-411, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[17] Paulo S.L.M. Barreto, and Vincent Rijmen, "The Khazad Legacy-level Block Cipher," *Primitive Submitted to NESSIE*, vol. 97, no. 106, pp. 1-20, 2000. [Google Scholar]

[18] Chang'e Dong, "Color Image Encryption Using One-Time Keys and Coupled Chaotic Systems," *Signal Processing Image Communications*, vol. 29, no.5, pp. 628-640, 2014. [CrossRef] [Google Scholar] [Publisher Link]