**How to Cite**:
Obaida, T. H., Jamil, A. S., & Hassan, N. F. (2022). Improvement of rabbit lightweight stream cipher for image encryption using Lévy flight. *International Journal of Health Sciences*, *6*(S8), 1628–1641. https://doi.org/10.53730/ijhs.v6nS8.11630

# Improvement of rabbit lightweight stream cipher for image encryption using Lévy flight

**Tameem Hameed Obaida**
Department of Computer Systems Techniques, Al-Furat Al-Awsat Technical University, Najaf Technical Institute, AL Najaf, Iraq
Corresponding author email: tameem.daham@atu.edu.iq

**Abeer Salim Jamil**
Department of Computer Technical Engineering, Al-Mansour University College, Iraq
Email: abeer.salim@muc.edu.iq

**Nidaa Flaih Hassan**
Department of Computer Science, University of Technology, Baghdad, Iraq
Email: 110020@uotechnology.edu.iq

***Abstract*---**The encryption of digital images plays a very important role in maintaining privacy and security, in order to protect the people, like witnesses. So, an individual image must be encrypted before it is sent over the Internet to the intended recipient. In this paper, images are encrypted using the rabbit algorithm and the chaotic equation represented by Lévy flight for several reasons, including employing beginning seed number, variance of parameters, and unpredictable random walk direction by making the encryption key more robust through the randomization of the initial vector values. After applying the original and improved rabbit encryption algorithms to the images, the results showed that the proposed Rabbit cipher has shown higher robustness against brute force assaults and larger capability of defending against the statistical cracking. Which has been verified and validated through the use of security metrics are PSNR, MSE, and entropy criteria., The encryption of a 256-size image was achieved in just 3 seconds when the improved Rabbit algorithm has been used, so it will be possible to encrypt video frames in real time because it requires a lesser time and having higher security.

***Keywords*---**image encryption, levey flay, rabbit, initial vector, Lévy flight.

## Introduction

One of the big concern in the field of digital image encryption is the ability to decrypt it by unauthorized persons[1, 2]. In-stream images, especially if the algorithm used is lightweight encryption Algorithm, because it uses simple arithmetic operations such as XOR, despite its speed, but it is characterized by low security[3].

Generally, the strength of the security provided by streaming encryption algorithms depends on the robust and length of the secret key. In order to increase security, these keys must be long enough (at least 128 bits) to meet basic security requirements [4]. The rabbit stream cipher algorithm was introduced as one of the Fast Software Encryption (FSE) in a 2003 workshop by Martin Boesgaard, Mette Vesterager, Thomas Christensen, and Erik Zenner[5]. Then an IV-setup function was designed and a security analysis was conducted, which at the time praised the lack of weaknesses of the algorithm when encrypting texts, because the rabbit algorithm was previously used to encrypt texts only [6]. The standard rabbit algorithm consists of a 128-bit Secret Key (SK) and a 64-bit Initial Vector (IV) as input, where a block of 128 pseudorandom bits are generated in each iteration, as a result of the process of merging the secret key bits with the initial vector and the output represent the value of the key used in the encryption [7, 8]. As for the size of the internal state, it consists of 513 bits divided into eight counters and eight state variables, each of which is 32 bits in size [9]. But it is worth noting that the value of the initial vector in the traditional algorithm is constant values. So, in order to have a more robust key, in this paper a new IV is used for Rabbit Stream by replacing it with the Lévy statistic. Related to chaos theory, which is derived from mathematics and is considered useful in providing randomness because it simulates the random nature. Which is later combine with the secret key for image encryption, the purpose of which is to extend the security levels, whereby secure keys can be generated effectively [7] [10]. The use of chaos theory is very important in the field of information security because many encryption algorithms depend on this principle in order to obtain high security, so the new model ensures the achieve of implementation and security requirements in addition to speed in encrypting digital images[11, 12] . The remainder of the paper is coordinated as follows: Section 2 presents related works. Section 3 portrays the layout of the proposed model. The execution and assessment of the proposed calculation is introduced in Section 4.

## Related works

There haven't been many applications that have used the Rabbit algorithm, in addition to the fact that most of them are used to encrypt texts only. Shweta et.al,[12] used model for design and simulation in order to secure data transmission using the rabbit algorithm for binary data encryption, which was implemented using Verilog and Modelsim 6.4a, where the results proved it as a safe, efficient and fast algorithm. Tahir et. Al,[13] presented a lightweight encryption method using a rabbit algorithm called LRSA. suggested a component utilized in remote detection organization to give classification, alluded to as LSRA i.e., lightweight encryption instrument dependent on Rabbit algorithm stream image. In remote sensor organizations, the LSR gives all security applications a

need for information secrecy. The LSR was executed to meet the objectives like execution, security, and convenience. Two plans were suggested, one is “Symmetric Key Cryptography” (SKC) based plan to encode mass information and another is “Public Key Cryptography” (PKC) to scramble a mysterious key utilized for correspondence. A test system of little OS i.e., TOSSIM was utilized to test LSRA. It takes 39 seconds to encode and unscramble 128 bits of plain text. Sreehari Kundella et.al,[14] proposed model is a lightweight cryptographic model which utilizes extremely restricted assets, shares exceptionally less memory, and the calculation just as the time taken is likewise exceptionally less contrasted with the other regular encryption plot. In this manner, the proposed MapReduce and Rabbit calculation gives security at a significant level against any sort of safety breaks. Leksono et.al,[15] suggested and dealt with an Email application utilizing the Rabbit calculation to get Email content in android advanced mobile phones. In this work encryption and decoding of email's substance was finished utilizing hare calculation. For this sort of utilization, they utilized java-script so it very well may be utilized to handle messages. Java provides a few free libraries for changing and supporting an application. The IMAP protocol is used to recover messages. To transmit messages, SMTP conventions are used. For the email application, Google mail is one of the email administrations. Yongsheng et.al, [16] suggested a method to develop a rabbit encryption algorithm to encrypt the signal bit of a motion vector, (IEMV), for H.264/AVC video encoding. Which has proven to be effective against attacks. Khaled Suwais et.al,[4] introduced an equal preparing model for multi core processors utilizing bunny stream images. The fundamental objective of this publication is to further develop a presentation so that a keystream age and encryption measures were sped up. Utilizing an equal model, the computational issue can be settled and by partitioning into fragments simultaneously. Sruthi L. et.al,[17] proposed a secure data transmission by using traditional rabbit cipher to get encrypted image by the RRBE framework. In order to hide the reverse data.

**The Proposed Model**

A Rabbit uses a key of 128 bit and an Initial Vector (IV) of 64-bit. In each iteration of the algorithm the encryption is 128 bits synchronously, resulting in an efficient bit stream cipher. The main stages of Rabbit's algorithm are: key setup, IV setup and encryption. Figure 1 shows the main steps of the original Rabbit encryption algorithm that was used to encrypt images. Figure 2 shows the improved Rabbit encryption algorithm. The proposed model has been tested on a computer having CPU cori7 with 4 gigabytes of RAM where the time elapsed was three seconds.

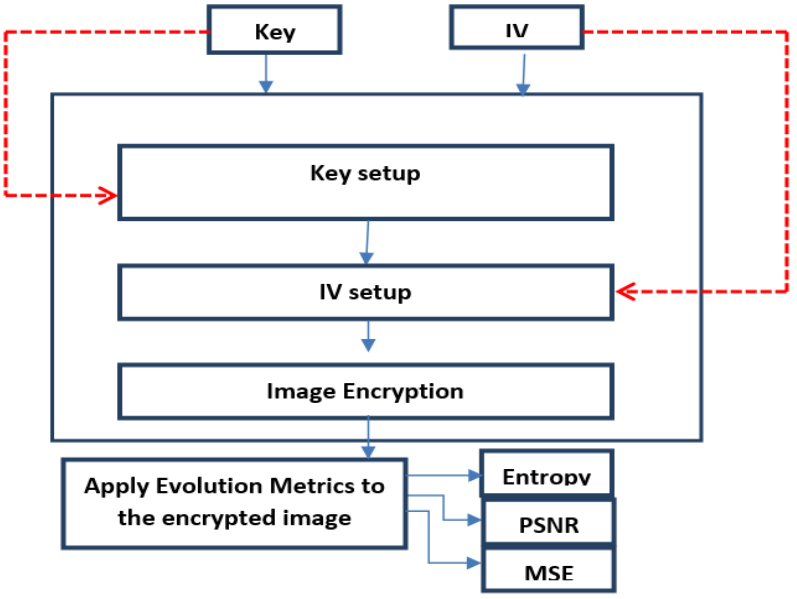



Figure 1: Original Rabbit Cipher for Image Encryption

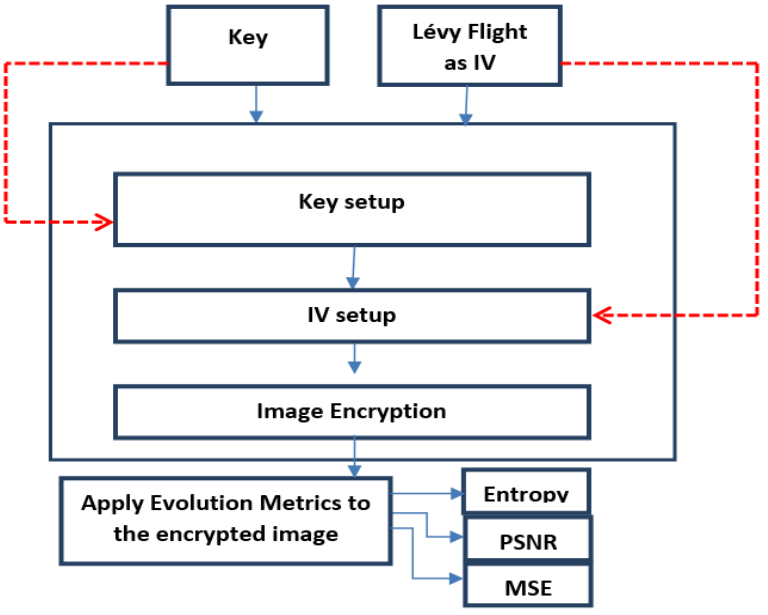



Figure 2: Proposed Rabbit Cipher for Image Encryption

**Key Setup**

The key is expanded as a first step to 128 bits divided into eight counters and state variables, each of the eight pieces contains two parts of the key, because the key bits are swapped by counters and state variables. Each piece contains 32 bits

[4]. These counters and state variables are configured by sub keys using the following Eq (1).

$$c_{j,4} = c_{j,4} \oplus x_{j+4 \bmod 8),4} \quad (1)$$

where $x_{j,0}$ is initial stat variables, and $c_{j,0}$ is initial counters for each $j$ to prevent recovery of the key by inversion of the counter system. The number 4 represents the number of iterations of the system used to reduce the correlation between the bits of the internal state variable and the key.

Figure 3 shows how to split and permutation the key bits of the state variables and counters, where the (x) represents the state variable and the (c) parameter represents the counter.

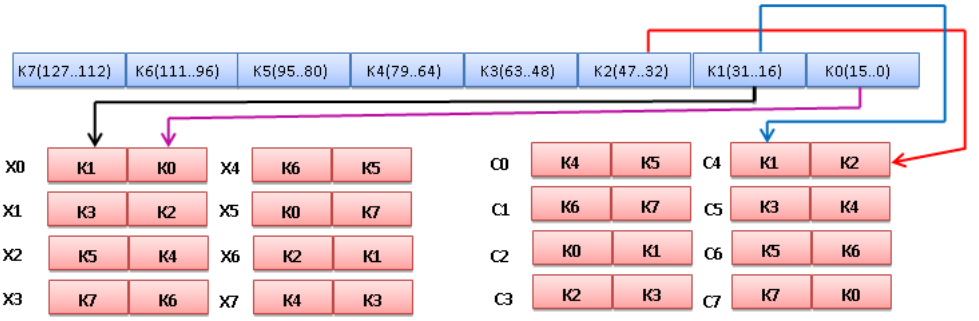



Figure 3: Shuffling the key bits where (x) is the state variable and (c) is the counter

**IV Setup**

The IV setup comes after the key setup scheme. By altering the counter function of IV [64…0], XOR the 64-bit initial vector on all 256 bits of the counter variables, IV setup is generated by Lévy flay. A Lévy flight is a random walk with a Lévy distribution, As mentioned earlier, Lévy flight generates random numbers because its idea stems from the chaotic theory derived from mathematics, so the constant values of IV in original algorithm, which can be easily reached, have been replaced by unpredictable random values, and Eq (2) A illustrates Lévy flight [18] [19][20].

$$Lévy = t - \lambda \quad (2)$$

where $\lambda$ is $(1 < \lambda \leq 3)$ and $t$ randomize initial number. Eq (2) will generate randomize numbers and these numbers will be converted to binary then used for IV [64…0] as shown in Eq (3).

$$c_{i,4} = c_{i,4} \oplus IV^{Lévy} \quad (3)$$

where $i \in \{0,1,2,3,4,5,6,7\}$ set number for eight steps. Here will generates eight sub-keys by make XOR IV and $c$.

**Image encryption**

In this step, the image is encrypted with Eq (4).

$$c_i = p_i \oplus s_i \tag{4}$$

where $c_i$ and $p_i$ denote i$^{th}$ cipher image. Also, $s_i$ is the 128-bit keystream block at iteration $i$ used for encryption and decryption. For decryption the image we used Eq (5).

$$p_i = c_i \oplus s_i \tag{5}$$

**Evaluation metrics**

This paper used three evaluation metrics to check the results, these metrics are peak signal to noise ratio (PSNR), mean square error (MSE), and entropy [21]. The PSNR computes the of twice images in decibels. This ratio is used to compare the quality of the original and compressed images. The PSNR increases as the quality of the compressed image improve. To compare image compression quality, the MSE and top PSNR are employed [22]. The MSE is a measure of peak error between the compressed and original image, whereas the PSNR is a measure of the cumulative squared error between a compressed and original image. The smaller the MSE number, the less the mistake. The PSNR is calculated by first calculating the mean-squared fault using the equation below:

$$MSE = \frac{\sum_{NM}[I_1(m,n) - I_2(m,n)]^2}{M * N} \tag{6}$$

M and N are the number of rows and columns in the input images, respectively, in the preceding equation. The PSNR is then calculated by the block using the calculation below:

$$PSNR = 10\ log_{10}\ (\frac{R^2}{MSE}) \tag{7}$$

The greatest fluctuation in the input image data type is represented by R in the previous equation. R is 1 if the incoming image is of the double-precision floating-point data type. R is 255 if the data type is an unsigned 8-bit integer, etc. The theoretical minimal average number of bits necessary to transmit a given source stream is known as entropy. The following Eq (8). may be used to calculate entropy, which is a highly significant quantity in images [23,24].

$$H = -\sum_k P_k\ log_2\ (P_k) \tag{8}$$

$P_k$ is the probability associated with gray level k, and k is the number of gray levels.

**Results**

Standard images (256*256) were used to test the proposed technique but it is possible to use images of different sizes with little difference in encryption time. The recommended Rabbit cipher image encryption based on Lévy flight was executed using Python 3.8. The key of 128 bit and IV of 64 bit is fed through the Rabbit algorithm, and the result is a random key of 128 bit. Putting them to use randomly generated keys, the plain image (256*256), is encrypted to get the cipher image. Finally, the encrypted image is XOR Rabbit key to the acquired original image. Figure 4 illustrates an original persona image.

Figure 4: Original Image

Figure 5 shows the histogram of the original image. So, the maximum intensity of all channels (red, green, and blue) is 250 and the maximum count of these intensities are 800 in general.

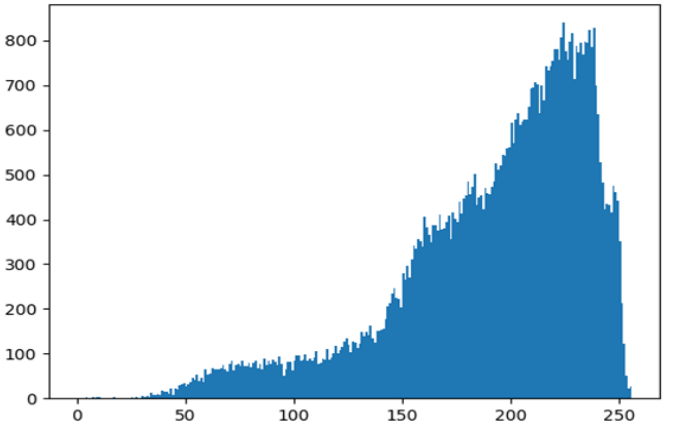



Figure 5: Histogram for the Original Image

Figure 6 shows more details for each channel, these channels are red, green, and blue. This distribution depicts the image have interfering colors.

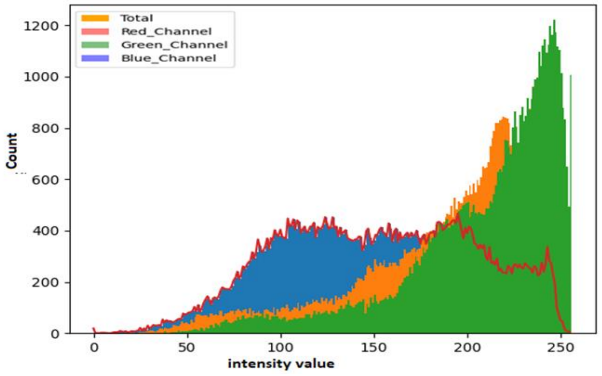



Figure 6: Chanel Color for the Original Image

Table 1 shows an example of how the key and IV are used to encrypt the original image. The difference is obvious between the values generated by Lévy flight and IV, where the Lévy flight generates a large number of random numbers and chooses part of them to generate the secret key instead of using simple and known IV values as in the original algorithm. So, eight sub-keys will be generated of random number by Lévy flight that presents in Eq (2).

Table 1. Key and IV for generate key stream

| Term | Value |
|---|---|
| Key | B4FdB42694892F5C1509BE952D2FBC1E |
| IV-generate by Lévy | 597E26C175F573C36D7D012292CC DC E0E21200 58B94ECD1F |
| IV choose | 6D58CD59DCE0C11F |
| Key stream | 7CE9D63A1004EC19A4BCAC70DEAE2618 |

Figure 7 shows the encrypting of the original image using the improved Rabbit algorithm. Where the randomness is increased and features of the image disappear. Also, it's getting harder to identify the image.

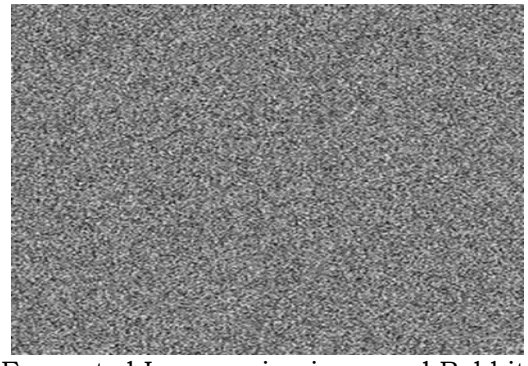


Figure 7: Encrypted Image using improved Rabbit algorithm

Figure 8 shows the histogram of the encrypted image with a size of 256 * 256 which can be compared with Figure 5.

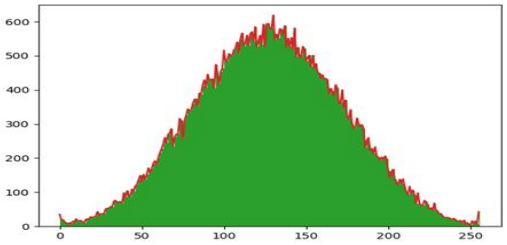



Figure 8: Histogram for Image Encryption

Table 2 shows the metrics that were applied to the image encrypted in Figure 7 and are PSNR, MSE and Entropy, respectively. where MSE and PSNR represent the encrypted image quality and Entropy measures the randomness of the encrypted image.

Table 2. Metrics applied to the encrypted image

| Metric | Value |
|---|---|
| PSNR | 27.906 |
| MSE | 105.306 |
| Entropy | 7.999 |

To compare the image encrypted by the standard Rabbit algorithm and improved Rabbit algorithm. In figure 9, (A) represents the image encrypted by the standard Rabbit algorithm in which the features of the image are rather clear, but in the image (B) encrypted by improved algorithm, it is impossible to identify the image, thus increasing the distortion and randomness of the encrypted image.

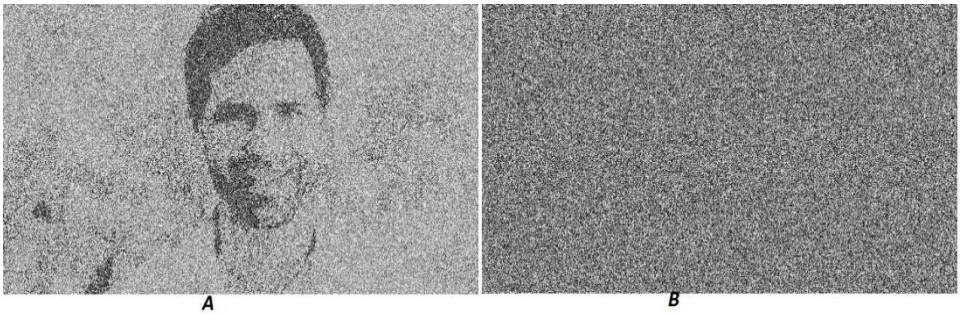



Figure 9: Comparison of the Encryption of the two Images (A) by Standard Algorithm(B) by Improved Algorithm

Table 3 show metrics comparison between image A and B. PSNR value (27.906) of encrypted image B is decreased; the lower quality of the encrypted image. MSE value (105.306) of encrypted image B increase; higher noise error. Thus, the quality and noise error are impact factors of increases the randomness of the encrypted image B where the Entropy value (7.999) increased of the encrypted image.

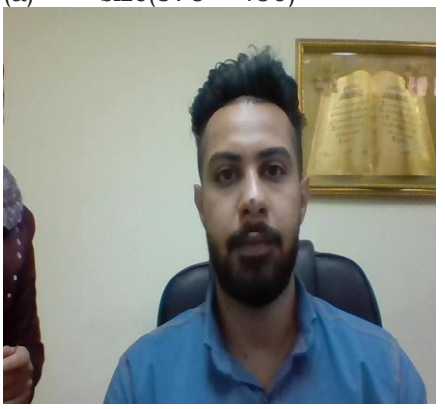
Table 3. Compare metric between images A and B

| Metric | Image A | Image B |
|---|---|---|
| PSNR | 28.179 | 27.906 |
| MSE | 98.873 | 105.306 |
| Entropy | 7.835 | 7.999 |

Figure 10 shows the histogram that was used to test the proposed technique by applying it to images of different sizes and obtaining the details of each channel, and these channels are red, green and blue. This distribution depicts the image in interfering colors.

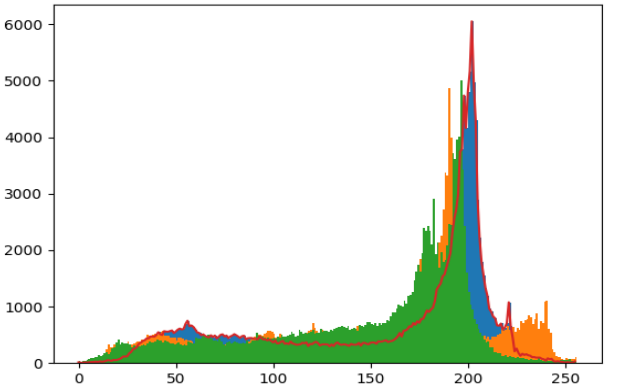
Original image

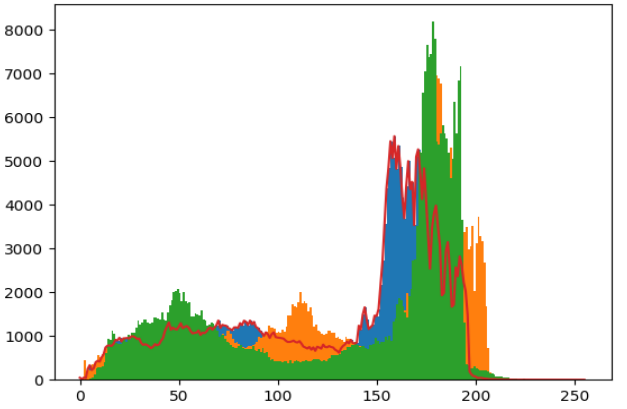
(a) size(376 × 430)

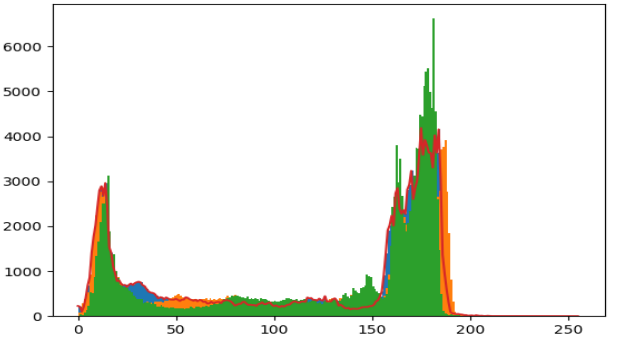


(b) size(640 × 480)

Histogram



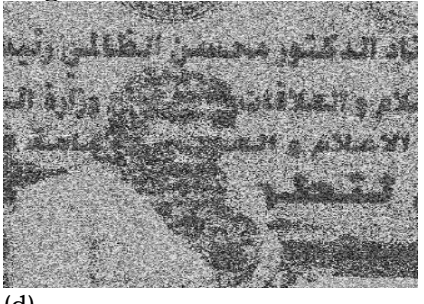



(c) size(493 × 343)

Figure 10: Original Images and Channel Color of each Image

Figure 11 shows a comparison between the original rabbit algorithm and improved algorithm, where images (d, e and f) represent the output of encrypting the original images (a, b and c) using the original algorithm, in which the features are clear, while images (g, h and i) are encrypted using the improved algorithm, in which image features disappeared due to increased randomness, which makes the improved algorithm more robust and secure.

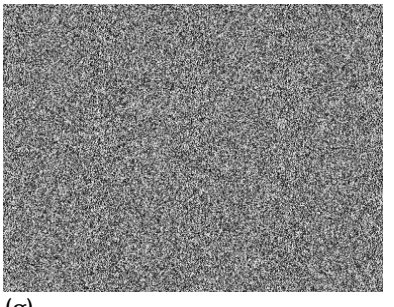
Original method



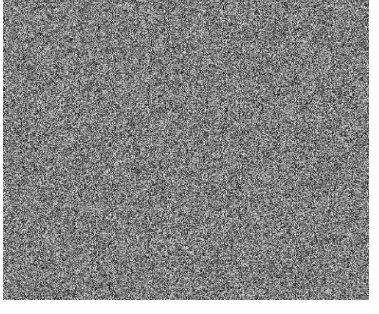
(d)

(e)

New method



(g)

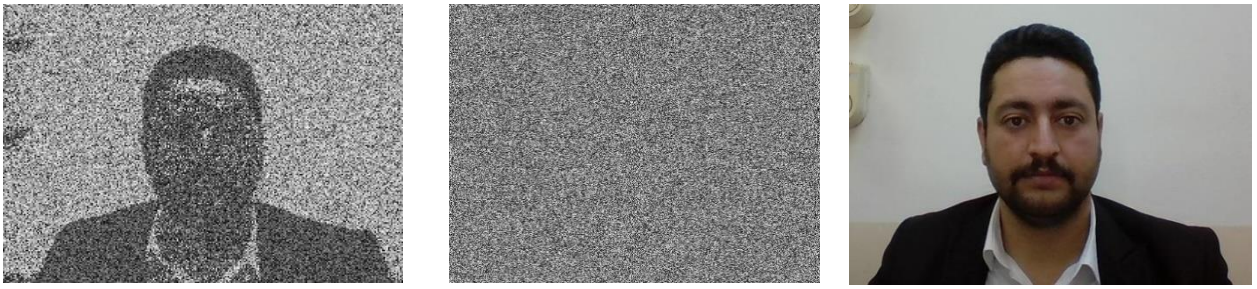


(h)

Decryption

(j)

(k)



(f) (i) (l)

Figure 11: Comparison of Images Encrypted by Standard and Improved Rabbit Algorithm

Table 4 shows the results of applying the metrics to the different images in Figure 11. Using the original algorithm, PSNR values for the images (d, e, and f) was (28.164, 28.189, 28.188) respectively, while in improved algorithm was (27.895,27.892,27.902), as well as increase in MSE values, that means ratio of noise and error is increased. When applying entropy metric to the images its selves, the results of the original algorithm was as follows (7.950,7.966,7.940), and improved algorithm was (7.998,7.999,7.998), which means that the randomness is increased and this is required in order to increase the security of the images.

Table 4. Comparison metric between the original and new method

| Image | Original Algorithm | | | New Algorithm | | |
|---|---|---|---|---|---|---|
| | Entropy | PSNR | MSE | Entropy | PSNR | MSE |
| Img_d | 7.950 | 28.164 | 99.228 | 7.998 | 27.895 | 105.559 |
| Img_e | 7.966 | 28.189 | 98.650 | 7.999 | 27.892 | 105.633 |
| Img_f | 7.940 | 28.188 | 98.676 | 7.998 | 27.902 | 105.400 |

**Conclusion**

Rabbit algorithm is fast and efficient, so it can be used to encrypt digital images, as is the case with encrypting texts, but after its practical applied to images, it turns out that the standard algorithm is inaccurate in encryption because the image can be distinguished easily, so this algorithm was improved by strengthening the key to increase Randomization using Lévy flights to produce random numbers for the initial vector, and thus the results proved that produced IV is more efficient than the standard Rabbit IV and according to the evaluation metrics used, Entropy, MSE, PSNR, there was a clear difference between the image encrypted using the original and improved algorithm for all images used, as shown in Table 4. For example, in image (d), the Entropy, MSE, and PSNR ratios were 7.950, 28.164, 99.228, respectively, using the original method, while the ratios were 7.998, 27.895, 105.559 using the proposed method. According to the proposed method, a brute force attack is impossible. It is also fast as it takes only 3 seconds to encrypt the image, so it can be used to encrypt video frames in real time. The encrypted file with the Lévy flight function has greater security than the encrypted file without Lévy flight, and the encrypted image by the proposed method has higher security than the original Rabbit algorithm.

**Reference**


1. Ail, Y.H. and Z.A. Alobaidy, Images encryption using chaos and random generation. Eng. &Tech. Journal, 2016. 34(1).
2. Akhyar, F., S.M. Nasution, and T.W. Purboyo. Rabbit algorithm for Video on Demand. in 2015 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob). 2015. IEEE.
3. Ali, N.H.M., A.M.S. Rahma, and A.S. Jamil, Text Hiding in Color Images Using the Secret Key Transformation Function in GF (2 n). Iraqi Journal of Science, 2015. 56(4B): p. 3240-3245.
4. Al-Janabi, S.T.F., K.S. Rijab, and A.M. Sagheer. Video Encryption Based on Special Huffman Coding and Rabbit Stream Cipher. in 2011 Developments in E-systems Engineering. 2011. IEEE.
5. Almheiri, A., et al., The entropy of Hawking radiation. Reviews of Modern Physics, 2021. 93(3): p. 035002.
6. Barthelemy, P., J. Bertolotti, and D.S. Wiersma, A Lévy flight for light. Nature, 2008. 453(7194): p. 495-498.
7. Boesgaard, M., et al. Rabbit: A new high-performance stream cipher. in International workshop on fast software encryption. 2003. Springer.
8. Chanana, M. (2018). Empirical study: relationship between self efficacy and academic performance. *International Journal of Health & Medical Sciences*, *1*(1), 28-34. https://doi.org/10.31295/ijhms.v1n1.36
9. Chechkin, A.V., et al., Introduction to the theory of Lévy flights. Anomalous transport, 2008. 129.
10. Choo, E., et al. SRMT: A lightweight encryption scheme for secure real-time multimedia transmission. in 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07). 2007. IEEE.
11. Feng, X., et al., On guess and determine analysis of Rabbit. International Journal of Foundations of Computer Science, 2011. 22(06): p. 1283-1296.
12. Fúster-Sabater, A., Computing Classes of Cryptographic Sequence Generators. Procedia Computer Science, 2013. 18: p. 2440-2443.
13. Gandara, R.B. and M. Alaydrus. Analysis of the IEEE 802.15. 4 Protocol with Rabbit Encryption Algorithm for Industrial Applications in Oil and Gas Sector. in 2019 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering. 2019. IEEE.
14. Kundella, S. and R. Gobinath, A Persuasive Rabbit Algorithm Enhanced with Map Reduce Security Mechanism for ECG Data Security in Cloud Storage. Annals of the Romanian Society for Cell Biology, 2021. 25(6): p. 8882-8893.
15. Leksono, M.A. and R. Munir. Email client application with rabbit algorithm for Android smart phone. in 2012 7th International Conference on Telecommunication Systems, Services, and Applications (TSSA). 2012. IEEE.
16. Mahdi, M. and N. Hassan, A suggested super salsa stream cipher. Iraqi Journal for Computers and Informatics, 2018. 44(2): p. 5-10.
17. Mahdi, M.S. and N.F. Hassan, Design of keystream Generator utilizing Firefly Algorithm. Journal of Al-Qadisiyah for computer science and mathematics, 2018. 10(3): p. Page 91-99.
18. Obaida, T. H., & Abd, D. H. (2016). A Robust Approach for Mixed Technique of Data Encryption Between DES and RC4 Algorithm. *Journal of Kufa for Mathematics and Computer*, *3*(2).

19. Prathima, N., S. Chetan, and S.M. Rehman. ASIC Implementation of Rabbit Stream Cipher Encryption for Data. in 2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE). 2019. IEEE.
20. Santra, S., et al. Real-Time Vehicle Detection from Captured Images. in 2019 International Conference on Opto-Electronics and Applied Optics (Optronix). 2019. IEEE.
21. Sharma, A., et al., Multi-level image thresholding based on Kapur and Tsallis entropy using firefly algorithm. Journal of Interdisciplinary Mathematics, 2020. 23(2): p. 563-571.
22. Sruthi, L., et al., Secure Data Transmission using Rabbit Stream Cipher. International Journal of Engineering Research, 2014. 3(4).
23. Suryasa, I. W., Rodríguez-Gámez, M., & Koldoris, T. (2021). Health and treatment of diabetes mellitus. *International Journal of Health Sciences*, *5*(1), i-v. https://doi.org/10.53730/ijhs.v5n1.2864
24. Suwais, K., Parallel Model for Rabbit Stream Cipher over Multi-core Processors. WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, 2014.
25. Tahir, R., et al. LRSA: lightweight Rabbit based security architecture for wireless sensor networks. in 2008 Second International Symposium on Intelligent Information Technology Application. 2008. IEEE.
26. Wang, Y., M. O'Neill, and F. Kurugollu. The improved sign bit encryption of motion vectors for H. 264/AVC. in 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO). 2012. IEEE.