# Modified Rabbit Encryption Algorithm With Novel 4-D Chaotic System

Hala Mohammad Yaroub [1] Jolan Rokan Naif [2] Shatha Mezher Hasan [3]
Iraqi Commission for Computer & *Informatics / Informatics Institute for Postgraduate Studies, Iraq* [1,2,3]

**ABSTRACT**
Inrecent years, the use of Wireless Sensor Networks (WSN) has become increasingly prevalent across various fields. This has led to agrowing need for secure data communication to protect sensitive information. To overcome the limitations of WSNs, lightweight algorithms have gained popularity due to their speed and efficiency.

This paper presents a novel solution for securing sensing data transmitted over WSNs. The proposed modified Rabbit encryption algorithm utilizes a 4-D chaotic system as a key generator to encrypt and decrypt data, providing a robust and efficient solution. The expanded key space generated by the 4-D chaotic system made brute force attacks practically impossible, ensuring data confidentiality and integrity. The use of a strong encryption algorithm and chaotic key generator enhances the system's security and efficiency. The modified Rabbit cipher algorithm encryption execution duration was less than the original Rabbit algorithm, and the chaos increased the randomness and complexity of the cipher. Overall, the proposed encryption algorithm provides a trustworthy solution for securing sensitive data in WSNs.

The results of the NIST and data quality tests suggest that the modified cryptographic Rabbit algorithm is more robust than the original Rabbit algorithm.
**Keywords**: Rabbit stream cipher, chaotic key generation, WSN security, IoT.

## 1. Introduction

In the past few decades, the electronic sharing of data has become increasingly prevalent due to advancements in data communications. However, sensitive information related to fields such as medicine and the military is often transmitted through unsecured communication channels, which poses a significant risk. With the open nature of the internet, safeguarding this data has become a major concern. Consequently, encryption was introduced as a solution to protect sensitive information from unauthorized access[1].

The use of IoT devices has led to an increased need for secure communication, resulting in the development of lightweight cryptography techniques. These techniques aim to improve performance while minimizing resource consumption. International Organization for Standardization(ISO) standards provide a foundation for lightweight encryption algorithms and key distribution methods can support security among nodes. While standard encryption and hashing algorithms are secure, they consume significant device resources, leading to a shift towards lightweight security[2].One of the essential challenges in encrypting data is the risk of unauthorized encryption, particularly when using a lightweight encryption algorithm. Such algorithms typically rely on simple arithmetic operations, like XOR, to achieve fast encryption, but this often results in a low level of security[3].

Encryptionalgorithms are usually assessed based on the strength and size of the secret key used, with larger keys generally leading to more secure encryption [4].

The Rabbit cipher algorithm, which is known for its high performance,was first introduced at the $10^{th}$ International Conference on Fast Software Encryption (FSE) in 2003[5].It is based on a symmetric key scheme and requires a 128-bit secret key and a 64-bit initialization vector (IV) as input. During the encryption process, the algorithm utilizes the internal state of 513 bits at each iteration to produce 128-bit pseudo-random bits that comprise the output block[6].

chaotic systems are currentlybeingemployed to enhance cryptosystems, and their properties such as sensitivity to initial values and boundedness make them well-suited for addressing security challenges in the IoT. As a result, they have become a popular choice for developing effective solutions to these issues[7].Researchers have successfully employed chaotic theory to encrypt extensive data sets, such as images, audio, and video data. This is because chaotic maps process valuable properties, including the ability to generate long-period keys, pseudo-random numbers, and sensitivity to changes in system parameters and initial conditions. However, despite the high security, high-speed data encryption, and low computational power and overhead requirements that come with using chaotic maps, there are still challenges that researchers must overcome. Consequently, there is a need for new methods that rely on the synergy between chaos and digital logic to encrypt information for fast and secure networks[8].The non-linear nature of chaotic systems makes them a potent tool for encryption, andresearchers have been driven to utilize chaos theory in cryptography because of the similarities between the two fields. Both chaos theoriesand cryptography share features, such as sensitivity to parameterchanges, long-term unpredictability, and random behavior [9].

R. H. Al-Hashemy and S. A. Mehdi introduced a new algorithm for image encryption that utilizes a novel 3D chaotic system based on magic squares. The proposed technique involves generating a set of chaotic keys using the chaotic system, arranging them into a matrix, and dividing them into non-overlapping submatrices. The original image is also divided into sub-images, each of which is multiplied by a magic matrix to produce another set of matrices. The XOR operation is then applied to the two sets of matrices to generate the encrypted image. The encryption method is evaluated in terms of security and statistical analysis and is found to be highly resistant to different types of attacks. Additionally, the encryption and decryption times are fast. Overall, the proposed encryption algorithm presents a simple yet effective solution for image encryption, utilizing a chaotic system and magic squares to achieve a high level of security [10].

H. K. Hoomod et al. proposes a secure IoT data sensing system based on a novel 5-D chaotic system and encryption algorithms, including a hybrid Speck-Present algorithm and modified Present and Speck algorithms. The proposed system provides high-level security for sensitive information generated from IoT sensors and is designed to provide users with flexibility and ease in managing change operations. The encryption algorithms passed several tests and are proven to have strong security due to a large key-space and passing all NIST tests[11].

The Rabbit algorithm has not been extensively utilized in practice, and the majority of its applications have been for text encryption. According to Obaida, T. H. et al., the Rabbit algorithm is a fast and efficient method for encrypting digital images, but it has been found to be inaccurate in practice. To enhance its efficiency, the algorithm has been modified to incorporate the use of the Lévy flight algorithm to produce random numbers for the initial vector, which has resulted in improved efficiency. The improved algorithm has been found to be more effective than the original algorithm in terms of various evaluation metrics, including entropy, MSE, and PSNR. Additionally, the improved algorithm is resistant to brute force attacks and can encrypt images in real-time, making it suitable for use in video applications. Overall, the use of the Lévy flights has resulted in enhanced security for images encrypted using the improved Rabbit algorithm[3].

V. Tiwari presents an optimized security protocol for WSNs that provides data confidentiality, authentication, and integrity while being energy-efficient. The protocol is designed using Rabbit stream cipher for confidentiality and Rabbit-based MAC function for authentication. The proposed protocol uses a new packet format that reduces packet size and promotes the reuse of fields in the IV, thus saving battery space. The protocol also uses the next state function for computing MAC, reducing the space requirement for maintaining a separate MAC algorithm[12].

## 2. The Proposed Cipher System

The proposed cipher system is an encryption scheme that combines the 4-D chaos keys generation stage and the modified Rabbit stream cipher to provide a highly secure and efficient data transmission as follows:

### 2.1. The 4-D Chaos Keys Generation Stage

Chaotic systems, due to their randomness, have gained popularity among researchers. Their outputs have been used in encryption operations in recent years. Several chaotic systems, such as the logistic, Lorenz, Hanon, Chen, and Cat systems, have been researched and used. The Lorenz system, for instance, has one positive dimension exponent, represented by Lyapunov (2.16, 0, -32.4). Several researchers have attempted to modify the Lorenz system to improve its Lyapunov exponent.

A novel 4-D chaos system was developed to meet the needs of the Rabbit algorithm, which required 4-D chaos keys (K1, K2, K3, and K4) for its security and complexity. The chaos key generator stage includes a novel chaotic system with different initial and parameter values, as shown in equation (1). The novel 4-D system Lyapunov exponent values were tested and produced Lyapunov exponent values of (1.008, -1.989, 0.564, and 0.332).

$$\left.\begin{aligned} xt[i+1] &= xt[i] - s*(xt[i]-yt[i]+kt[i]*kt[i])*dt \\ yt[i+1] &= yt[i] + (-yt[i]-xt[i]*zt[i]+r*xt[i]+kt[i])*dt \\ zt[i+1] &= zt[i] + (xt[i]*yt[i]-b*zt[i]+kt[i])*dt \\ kt[i+1] &= kt[i] + (u * yt[i] * (1-kt[i]))*dt \end{aligned}\right\} \quad \ldots(1)$$

Where b=8.8/3.0,r=30.0, s=11.0, and u=0.25are the chaos parameters, while x=0.400000000001, y=0.10002, z=0.1903, and k=0.102 are the initial values of the chaotic system.

These equations were used to generate dynamic keys (chaos generation keys (K1, K2, K3, and K4)), which are sensitive to the initial values, and any slight change in the initial values will lead to a significant change in the output value. To ensure more complexity, the new chaos keys were generated from the four-dimensional equations of the novel chaotic system using K1 for the 128-bit secret key and K2 for a 64-bit IV. The map of the proposed novel 4-D Chaotic System is shown in Figure 1. Algorithm (1) shows the generation of the chaos key operation.

**Algorithm (1): The 4-D Chaotic System Algorithm**

**input:** initial values and parameters. where x = 0.400000000001, y = 0.10002, z = 0.1903, k = 0.102, and d(t) = 0.01 are the initial values, while b = 8.8/3.0, r = 30.0, s=11.0, and u = 0.25 are the chaos parameters.
**Output:** Chaos Keys K1, K2, K3, and K4.

**Begin**
**Step 1:**Calculate the values of xt, yt, zt, and kt using the novel 4-D chaotic system equations(1).
**Step 2:** Split xt[i+1], yt[i+1], z[i+1], and k[i+1] into fractional part(v1, v2, v3, and v4) and integer part(n1, n2, n3, and n4) respectively.
**Step 3:**Convert the (v1, v2, v3, and v4) to positive integer numbers and split each of them into two parts fractional v11, v21, v31, and v41 and integers n11, n21, n31, and n41 (which represent chaos keys K1, K2, K3, and K4) respectively.
**Step 4:**Save the values of keys K1, K2, K3, and K4 to a file.
**End**.

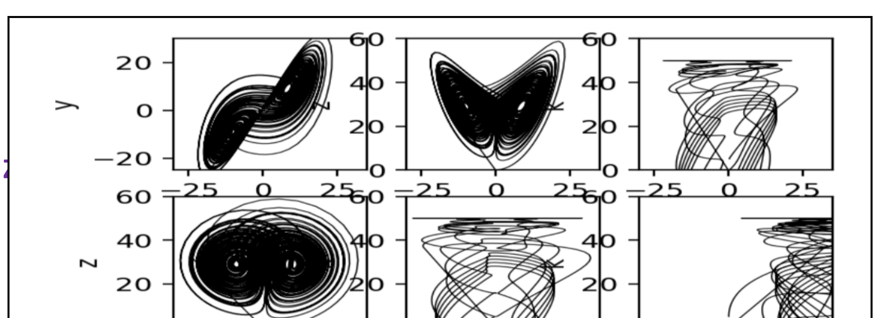
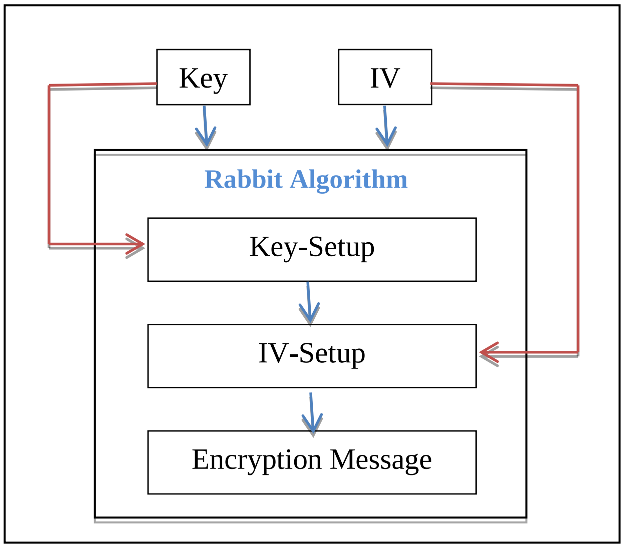


Figure 1: Map Of the Proposed Novel 4-D Chaotic System.

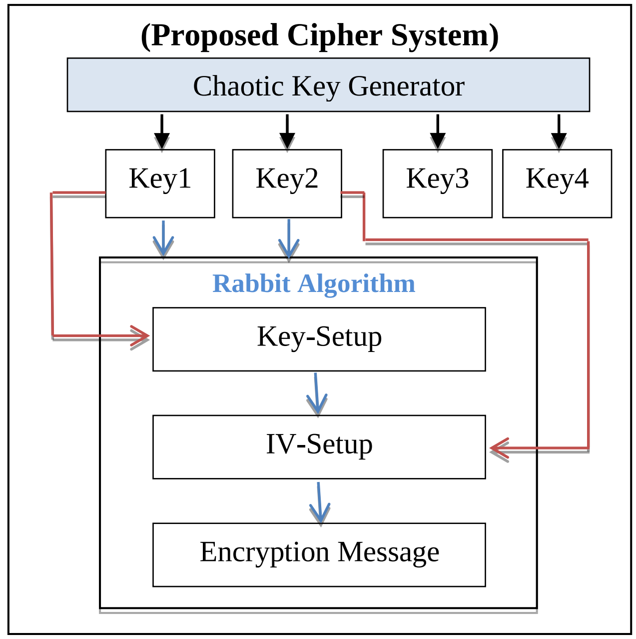
### 2.2. The Modified Rabbit Stream Cipher

Rabbit is a high-performance stream cipher algorithm that utilizes the same secret key for encrypting and decrypting data. The algorithm takes a 128-bit key and a 64-bit IV as inputs andproduces a block of 128 pseudo-random bits after each iteration by combining the internal state 513 bits, and then performs encryption and decryption by XORing the resulting pseudo-random data with plaintext or ciphertext.The internal bits are split into eight state variables and eight counter variables, with each variable being 32 bits in length, along with one counter carry bit that is necessary to be stored between iterations. With a 128-bit key, there are $2^{128}$ possible key combinations, which is a very large number. This makes it extremely difficult, if not impossible, for an attacker to try all possible keys and successfully decrypt the ciphertext [13].

The Rabbit algorithm was modified to work in conjunction with the Novel 4-D Chaos System that generates chaos keys (K1, K2, K3, and K4). K1 is used as the encryption key and K2 is used as the IV. This occurs to both parties (encryption/decryption).To increase security, the chaos keys may change over time, so K3 and K4 are used instead of K1 and K2 at different times. Figure 2 shows the original and modified Rabbit algorithms and the difference between them.




Figure(2-a): Original Rabbit Algorithm [3].




Figure(2-b): Modified Rabbit Algorithm.

Figure 2: The block diagrams show the difference between the original and the proposed rabbit stream cipher.

## 3. The Proposed System Implementations and Results

The proposed cipher system consists of two stages: chaos key generation and encryption. In this section, we will discuss each stage of implementation along with the corresponding results.

### 3.1. Chaos Keys Generator

A chaotic system is used to generate four-dimensional chaotic keys. The system is a modified version of the Lorenz system that produces four random sequences, namely K1, K2, K3, and K4. The chaotic nature of the system was verified by the presence of positive Lyapunov exponent values. This approach ensures that the generated keys are highly unpredictable and suitable for cryptographic purposes.

Figure 3 shows the results of the chaos values when applying the novel 4-D Chaotic System. While Figure 4 shows the significant difference observed in the results when there is a slight modification in the initial values of the system, that is, $Y_0$, when compared.

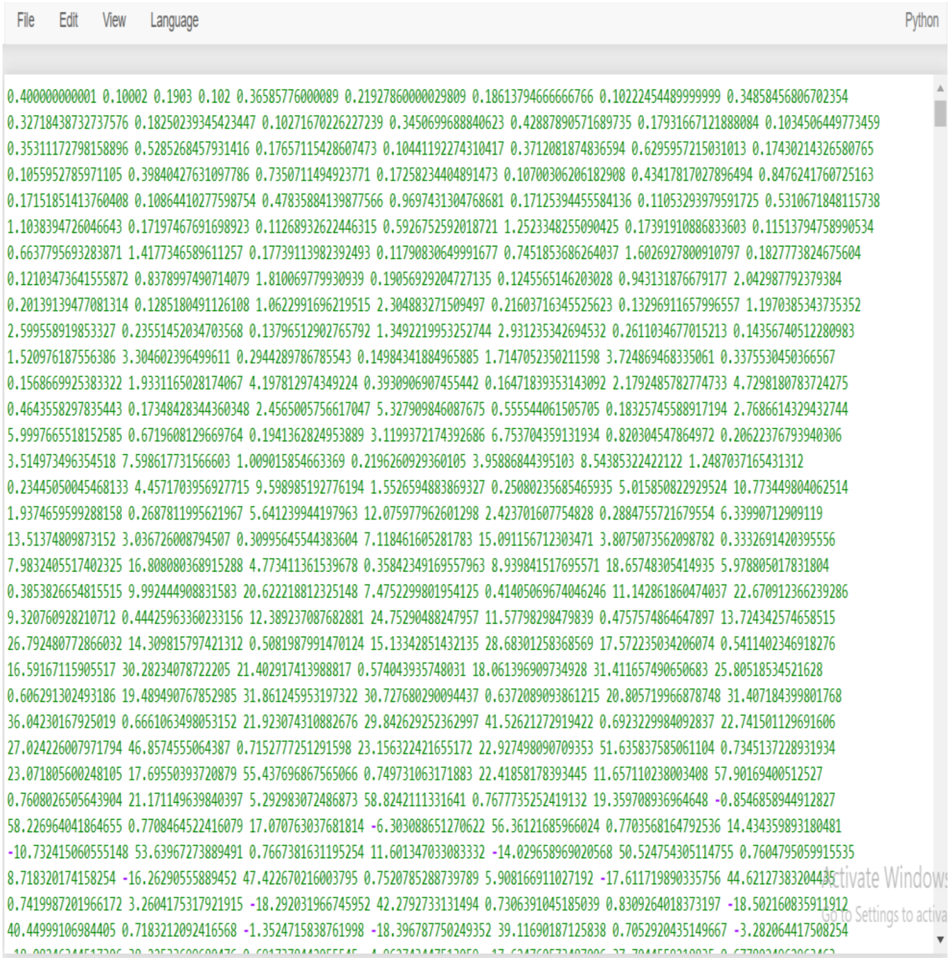



Figure 3: Sample of the result of apply the Novel 4-D Chaotic System ($X_0$ = 0.400000000001, $Y_0$ = 0.10002, $Z_0$ = 0.1903, $K_0$ = 0.102)

ISSN NO: 2230-5807

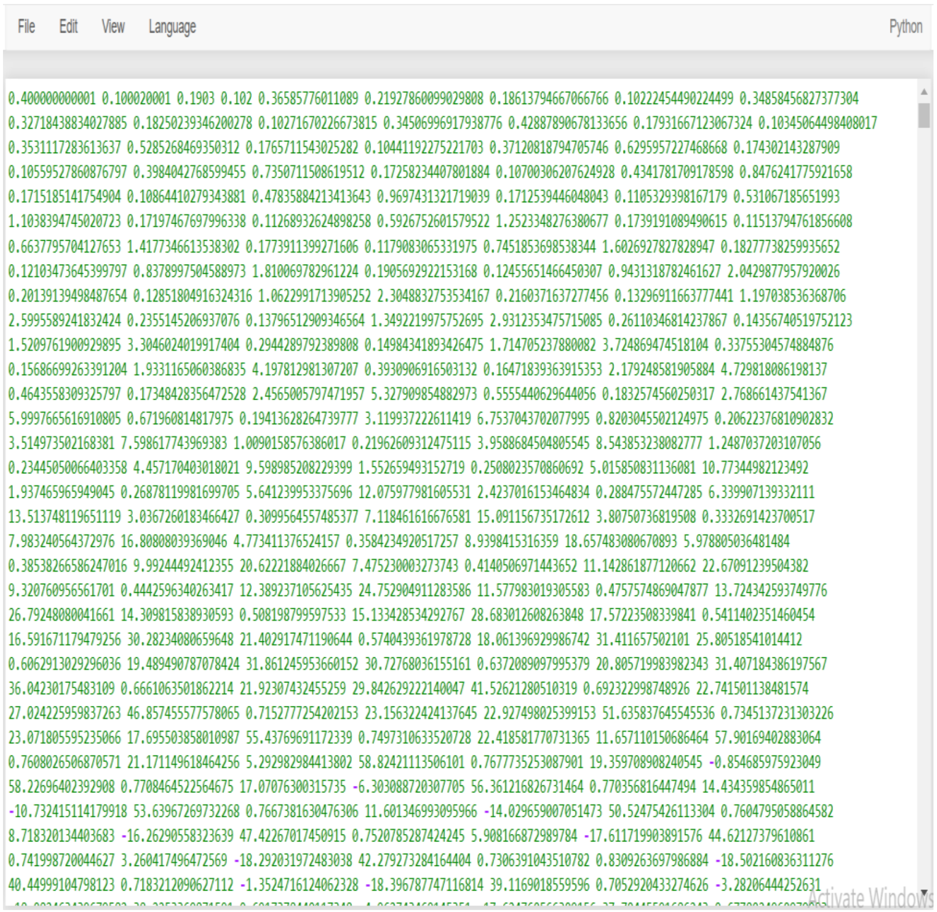


Figure 4: Sample of the result of apply the Novel 4-D Chaotic System
($X_0$ = 0.400000000001, $Y_0$ = 0.100020001, $Z_0$ = 0.1903, $K_0$ = 0.102)

The variation between the results of Figures (3) and (4) indicates a slight increase in one of the initial values, not exceeding 0.000000001. In Figure 3, the initial values used were ($X_0$ = 0.400000000001, $Y_0$ = 0.10002, $Z_0$ = 0.1903, and $K_0$ = 0.102), whereas the results shown in Figure 4 were obtained using the initial values ($X_0$ = 0.400000000001, $Y_0$ = 0.100020001, $Z_0$ = 0.1903, and $K_0$ = 0.102).

The system generates four chaotic keys as the final result. The decimal part of the floating-point value of xt[i+1] is first multiplied by 100000, and then passed to a function that returns the fractional and integer parts. The fractional part is multiplied by 10000 and passed back to the function to obtain the fractional and integer parts again. The resulting fractional part (a decimal value between 0 and 1) is then converted to a hexadecimal number using the hex function and stored as a string. This process is repeated for xt, yt, zt, and kt to get a unique identifier for each point in space. The results are stored in a file and used for the encryption and decryption phases that occur in the system that will use the proposed mechanism.

### 3.2. Key Sensitivity Analysis

To evaluate the sensitivity of the cipher system's key, one of the initial values of the proposed chaos key, $Y_0$, was changed slightly from 0.10002 to 0.100020001. This small difference in the input initial value resulted in a significant difference in the output generated by the 4-D chaos keys, as shown in Figure 4.

As a result, a slight change in the key value when applying the same text leads to a significant difference in the encoded text, showing the key's sensitivity. This is due to the algorithm's sensitivity to changes that may occur in the key, as illustrated in Table 1.

Table 1: Key Sensitivity with Modified Rabbit Algorithm

| | |
|---|---|
| Input Text | Iraqi Commission for Computers and Informatics/ Informatics Institute for Postgraduate Studies |
| Key 1 (chaotic) | 1e5011d826f06bf1ffb10b41fea22891 |
| IV 1 (chaotic) | 2198112322c91abd |
| Encrypted Text | c2bac28b1d4fc3b66fc3a4c2bcc28419536e7d6e6c6fc29b3b41c3a56b667d64c3b4c2b715c395c2aa1fc296c2bac394c2b94e7ec3b5c2abc28901c3943d5ac3be28c396c3bdc2a0547b61c3b5c2ae0044c3b1c2af15c2931042c3a53f4cc39cc2a1c29e101a7bc3a1c2b5c383c2a1174f6ac3a8c2b5024a62c3aa31c387717b607f616b6473 |
| Key 2 (chaotic) | 1e5011d826f16c01ffb10b41fea228a1 |
| IV 2 (chaotic) | 2198112422ca1abe |
| Encryp | 35c38c3ec39e3e0bc39625c388c2bf0047c3a924c3893dc3bf09c39829c28d154478c3bac2b0c296144ac3afc3ae06c3a521c282c29806c392c2b5c29f1bc39ac2a9c287c29408c29217c2b009c39829c38037c39fc2bc09c3867a64787871c3abc3a6c2bcc290c297595a71c3bdc3a733c39ec2abc29811c3893c4cc3a1c |

| | |
|---|---|
| te d Te xt | 2ab11c397c3b9c2bf02c38ec2b907c39228 |

### 3.3. Key Space

The modified Rabbit encryption algorithm utilizes a 128-bit key, providing $2^{128}$ potential keys. However, by incorporating the new 4-D chaotic system, the algorithm generates an infinite number of chaotic keys, significantly expanding the key space and increasing the difficulty for attackers attempting to perform brute force attacks to guess the secret key or IV. The size of the key space is calculated from the 4-D chaos key generation's parameters and initial values ($X_0$, $Y_0$, $Z_0$, $K_0$, b, r, s, u), resulting in a key space size that effectively resists brute force attacks. This robustness makes the proposed algorithm a trustworthy solution for encryption and decryption operations.

### 3.4. NIST Statistical and Other Standard Tests

The proposed cipher algorithm underwent a series of tests using the National Institute of Standards and Technology (NIST) suite, which is widely recognized for evaluating cryptography system security. The objective of these tests was to evaluate the strengths and potential weaknesses of the proposed algorithm. The results of these tests were analyzed in detail and presented for the modified Rabbit cipher algorithm. Table 2 provides a summary of the NIST suite test results conducted on the modified algorithm, confirming that the proposed modifications offer robust security and can effectively defend against various attack types. A test is considered successful if the p-value falls within the range of 0.01 and 0.99, indicating a high level of confidence in the cipher's security.

Table 2: The NIST Tests Results of the Modified Rabbit Algorithm

| Test no. | NIST statistical tests Results Name | P-value |
|---|---|---|
| **Test 1** | Frequency Test | 0.6211 |
| **Test 2** | Frequency within Block Test | 0.6834 |
| **Test 3** | Run Test | 0.5158 |
| **Test 4** | Longest-Run-of-Ones in a Block Test | 0.7894 |
| **Test 5** | Binary Matrix Rank Test | 0.5180 |
| **Test 6** | Discrete Fourier Transform Test | 0.4286 |
| **Test 7** | Non-Overlapping Template Matching | 0.5609 |
| **Test 8** | Overlapping Template Matching Test | 0.3690 |
| **Test 9** | Maurer's Universal Statistical | 0.9796 |
| **Test 10** | Linear Complexity | 0.6793 |
| **Test 11** | Serial Test | 0.4894 |
| **Test 12** | Approximate Entropy | 0.6003 |
| **Test 13** | The Cumulative Sums test | 0.5472 |
| **Test 14** | Random Excursions Test | 0.3812 |
| **Test 15** | Random Excursions Variant | 0.8942 |

Table 3 presents the encryption and decryption times for the modified Rabbit and Rabbit algorithms when operating on files of varying sizes. The results of the experiments demonstrate that, overall, the modified Rabbit algorithm is faster than the Rabbit algorithm in terms of both encryption and decryption. While the time difference in performance between the two algorithms is generaly small, it becomes more significant with larger files. The table indicates that the modified Rabbit algorithm is ultra-lightweight.

Table 3:Modified Rabbit and Rabbit algorithms encryption/decryption results for various sizes of files

| File Size bytes | Modified Rabbit Enc/Dec Time (msec) | Rabbit Enc/Dec Time (msec) |
|---|---|---|
| 128 | 6.081 | 6.394 |
| | 6.021 | 6.424 |
| 256 | 9.804 | 9.917 |
| | 9.726 | 9.907 |
| 512 | 21.013 | 23.128 |
| | 21.009 | 23.108 |
| 1024 | 34.984 | 38.864 |
| | 34.948 | 38.792 |
| 2K | 53.832 | 62.945 |
| | 52.986 | 62.901 |
| 4K | 84.446 | 98.098 |
| | 83.096 | 97.059 |
| 8K | 133.997 | 177.811 |
| | 131.251 | 176.642 |
| 16K | 211.057 | 279.308 |
| | 209.253 | 277.036 |
| 32K | 346.217 | 468.223 |
| | 338.988 | 464.981 |
| 64K | 590.106 | 797.708 |
| | 579.892 | 789.265 |
| 128K | 785.799 | 1431.750 |
| | 780.279 | 1428.318 |
| 256K | 1797.997 | 2353.910 |
| | 1788.102 | 2349.078 |
| 512K | 3898.267 | 4547.846 |
| | 3887.887 | 4540.938 |
| 1M | 7689.777 | 9237.144 |
| | 7680.099 | 9230.285 |

The result of the Hamming Distance test for the modified Rabbit and Rabbit encryption algorithms is shown in Table 4, where the test was conducted on texts of various sizes measured in bytes. The data in the table reveals that, in general, the Modified Rabbit algorithm produces a higher Hamming Distance compared to the Rabbit algorithm. This implies that the Modified Rabbit algorithm generates encrypted messages that are more dissimilar to the original message compared to the Rabbit algorithm. This suggests that the Modified Rabbit algorithm may offer better security as it is more challenging to derive the original message from the encrypted message.

Table 4: Hamming Distance Results of Encryption for Modified Rabbit and Rabbit Algorithms on Data of Different Sizes

| **Text Size (byte)** | Modified Rabbit | Rabbit |
|---|---|---|
| 128 | 447 | 310 |
| 256 | 927 | 735 |
| 512 | 1897 | 1477 |
| 1024 | 3698 | 2967 |
| 2048 | 7385 | 6198 |

Table 5presents the results of a data quality test for Modified Rabbit and Rabbit encryption algorithms using various byte sizes of data. The table presents four metrics for evaluating the quality of the encrypted data: Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Signal-to-Noise Ratio (SNR), and Entropy.

Upon examining the results presented in the table, it is evident that the Modified Rabbit algorithm generally outperforms the Rabbit algorithm on all byte sizes. Specifically, for all byte sizes tested, the Modified Rabbit algorithm has higher MES and Entropy values, indicating better quality. Moreover, the PSNR and SNR values of the Modified Rabbit algorithm are lower than those of the Rabbit algorithm, indicating better quality in terms of the dissimilarity between the original data and the encrypted data.

Table 5: Encryption data quality test for Modified Rabbit and Rabbit Algorithms using Different Byte Sizes of Data

| Byte | Encryption alg. | MSE | PSNR | SNR | Entropy |
|---|---|---|---|---|---|
| 128 | Modified Rabbit | 997.33 | 0.412 | 1.686 | 3.816 |
| | Rabbit | 946.16 | 0.533 | 1.895 | 3.522 |
| 256 | Modified Rabbit | 998.88 | 0.698 | 1.597 | 3.698 |
| | Rabbit | 896.84 | 0.491 | 1.789 | 3.475 |
| 1024 | Modified Rabbit | 988.11 | 0.726 | 1.483 | 3.803 |

| | | | | | |
|---|---|---|---|---|---|
| | Rabbit | 886.40 | 0.634 | 1.899 | 3.521 |
| 2048 | Modified Rabbit | 1110.23 | 0.579 | 1.301 | 3.786 |
| | Rabbit | 868.05 | 0.628 | 1.998 | 3.450 |
| 1M | Modified Rabbit | 1207.88 | 0.889 | 1.334 | 3.779 |
| | Rabbit | 838.98 | 0.810 | 2.412 | 3.598 |

## 3. Conclusion

The proposed algorithm offers a highly effective and efficient approach to safeguarding data transmitted over WSNs. The Modified Rabbit encryption algorithm, which employs a 4-D chaotic system as a key generator, presents a robust and efficient solution for encrypting and decrypting data. The algorithm's ability to resist attacks from malicious entities attempting to break the encryption is demonstrated by its sensitivity to minor changes in the key value. Additionally, the 4-D chaotic system generates an expanded key space that renders brute force attacks practically infeasible, thereby guaranteeing the confidentiality and integrity of the data. The recorded encryption and decryption time of the data indicate that the algorithm is well-suited for real-time applications. The utilization of a powerful encryption algorithm and the chaotic key generator enhances the security and efficiency of the system. Notably, the modified Rabbit cipher algorithm's encryption execution duration was lower than that of the original Rabbit algorithm, and the inclusion of chaos increased the randomness and complexity of the cipher algorithm. Ultimately, the proposed encryption algorithm offers a reliable solution for securing sensitive data in WSNs.

The proposed encryption algorithm passed all the NIST and other standard tests, indicating that it is highly secure and suitable for practical use.

## References

[1] A. Qayyum *et al.*, "Chaos-Based Confusion and Diffusion of Image Pixels Using Dynamic Substitution," *IEEE Access*, vol. 8, no. July, pp. 140876–140895, 2020, doi: 10.1109/ACCESS.2020.3012912.

[2] J. R. Naif, G. H. Abdul-Majeed, and A. K. Farhan, "Secure IOT System Based on Chaos-Modified Lightweight AES," *2019 Int. Conf. Adv. Sci. Eng. ICOASE 2019*, pp. 12–17, 2019, doi: 10.1109/ICOASE.2019.8723807.

[3] T. H. Obaida, A. S. Jamil, and N. F. Hassan, "Improvement of rabbit lightweight stream cipher for image encryption using Lévy flight," *Int. J. Health Sci. (Qassim).*, no. August, pp. 1628–1641, 2022, doi: 10.53730/ijhs.v6ns8.11630.

[4] A. Nanda, D. Puthal, S. P. Mohanty, and U. Choppali, "A Computing Perspective of Quantum Cryptography [Energy and Security]," *IEEE Consum. Electron. Mag.*, vol. 7, no. 6, pp. 57–59, 2018, doi: 10.1109/MCE.2018.2851741.

[5] I. D. Kuznetsov, Alexandr A. and Gorbenko, *Isci'2019: Information Security in Critical Infrastructures*. 2019.

[6] F. Akhyar, S. M. Nasution, and T. W. Purboyo, "Rabbit algorithm for Video on Demand," *APWiMob 2015 - IEEE Asia Pacific Conf. Wirel. Mob.*, pp. 208–213, 2016, doi: 10.1109/APWiMob.2015.7374978.

[7] Q. Shen, W. Liu, Y. Lin, and Y. Zhu, "Designing an image encryption scheme based on compressive sensing and non-uniform quantization for wireless visual sensor networks," *Sensors (Switzerland)*, vol. 19, no. 14, 2019, doi: 10.3390/s19143081.

[8] M. K. Ibrahem, "Perfect Secrecy System Based on Chaotic Key Generator," *Iraqi J. Inf. Commun.*

*Technol.*, vol. 1, no. 2, pp. 1–12, 2018, doi: 10.31987/ijict.1.2.4.

[9] N. M. Naser and J. R. Naif, "NEW ULTRA-LIGHTWEIGHT IoT ENCRYPTION ALGORITHM USING NOVEL CHAOTIC SYSTEM," *Int. J. Tech. Phys. Probl. Eng.*, vol. 14, no. 4, pp. 253–259, 2022.

[10] R. H. Al-Hashemy and S. A. Mehdi, "A new algorithm based on magic square and a novel chaotic system for image encryption," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1202–1215, 2020, doi: 10.1515/jisys-2018-0404.

[11] H. K. Hoomod, J. R. Naif, and I. S. Ahmed, "A new intelligent hybrid encryption algorithm for IoT data based on modified PRESENT-Speck and novel 5D chaotic system," *Period. Eng. Nat. Sci.*, vol. 8, no. 4, pp. 2333–2345, 2020, doi: 10.21533/pen.v8i4.1738.

[12] V. Tiwari, "Data Integrity and Authentication in WSNs," *HMR Interdiscip. J. Sci. Technol. Educ. Manag.*, vol. 4, no. 1, pp. 12–17, 2018, [Online]. Available: https://www.hmritm.ac.in/wp-content/uploads/2020/04/HMR-Journal-vol-4-issue1.pdf#page=17.

[13] V. Aswini, "A Structured Encrypted Authentication For Wireless Sensor Networks," vol. 19, no. 3, pp. 4346–4356, 2020, doi: 10.17051/ilkonline.2020.03.735586.