Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 2: 2025
ISSN : **1906-9685**

HYBRID ALGORITHM FOR SECURE AND EFFICIENT IMAGE, TEXT ENCRYPTION IN IOT SYSTEMS

[1]SAI MAHESH YERRAMSETTI, [2]RAYAPUDI BHUVANESWARI, [3]TUMURUGOTI RUSHITHA, [4]BARMAVATH SHIVA SAI, [5]KOGANTI VAHINI CHOWDARY

[12345] Department of ECE, Anurag Engineering College, Kodad Anantagiri Suryapet (Dist), Telangana, India.
[1]saimaheshyerramsetti@gmail.com, [2]bhuvana.rayapudi111@gmail.com, [3]rushitha64@gmail.com
siddubarmavath09@gmail.com, [5]vahinichowdary81@gmail.com

Abstract— Data security and confidentiality are essential for safeguarding user privacy, particularly in edge-fog-cloud systems and IoT devices that require quick, accurate answers yet are vulnerable to hacking. Strong security is provided by encryption techniques like Advanced Encryption Standard (AES), although they frequently have substantial computing costs. The enhanced AES technique presented in this study is perfect for encrypting messages and images in Internet of Things networks because it maximizes security while reducing computational cost. Our approach presents a two-stage key generation procedure: a three-dimensional Lorenzo function is introduced after an inventive chaotic function with great variable sensitivity. In contrast to conventional AES, we ensure fast encryption times of less than a millisecond by using distinct dynamic substitution boxes for odd and even rounds and keeping a single key for all rounds. A bit-level cyclic permutation takes the place of the mix column function, improving performance and speed. According to simulations and NIST benchmark testing, our technique significantly increases encryption speed, unpredictability, and efficiency, demonstrating its suitability for strong IoT data security.

Index Terms—AES Advanced Encryption Standard, edge-fog-cloud, NIST.

I. INTRODUCTION

By allowing linked objects to exchange data and communicate with one another, the Internet of Things (IoT) has transformed a number of industries, smart homes, and healthcare. Significant security difficulties have been brought about by this change, though, especially for low-resource devices that would find it difficult to put strong security measures in place. [1] By processing data closer to its source, fog computing can improve data security and privacy, but it also uses a lot of processing power, which sometimes makes it inappropriate for contexts with limited resources. The creation of lightweight cryptographic algorithms that can guarantee data secrecy and integrity without sacrificing these devices' performance is therefore required.

Many people value the Advanced Encryption Standard (AES) for its dependability and versatility in a range of applications. However, low-powered IoT devices may find it difficult to manage typical encryption algorithms due to their complexity. A good substitute that strikes a balance between security and resource efficiency is lightweight cryptography [2]. These methods are intended to enable safe data transmission and storage while preserving the pace at which devices with restricted resources can function. on this work, a novel lightweight encryption technique designed for text and color and grayscale images on resource-constrained IoT networks is presented. The main objective is to improve security in a fog computing environment by combining improvements to the AES algorithm with chaos-based key [3,4] generation. Implementing chaos-based key generation, which uses chaotic maps to create pseudo-random key streams that are extremely sensitive to beginning conditions and impervious to statistical attacks, is one of the research's main accomplishments. This innovative method shows a lot of potential for protecting image and text data in Internet of Things applications.

The suggested IoT AES upgrade further optimizes the AES algorithm to boost picture encryption performance on low-resource devices while preserving a high degree of security. Simulations show that the suggested approach maintains security while achieving remarkable encryption and decryption throughputs of 32,803.69 and 92,619.39, respectively [5] Moreover, the significant time and resources needed to crack the updated AES algorithm demonstrate its resilience.

This is how the paper is organized: A comprehensive assessment of pertinent literature is given in Section II, the suggested methodology is described in Section III, the results and comparative analysis are presented in Section IV, and the study is concluded and discussed in Sections V and VI. This all-inclusive strategy seeks to progress IoT [6] security by

providing practical answers to the problems presented by low-resource settings.

II. RELATED WORK

This section examines the evolution of lightweight picture encryption methods and secure key generation.

In order to increase security and resistance to attacks, researchers in [7] integrated the Henon map with the AES algorithm. The Henon map generated random keys for the encryption stage. The results indicated that the picture encryption performance was satisfactory. An AES-based method improved S-box production in [8], demonstrating an average avalanche impact of 51%, which was 3% greater than the first AES. Stronger encryption resilience was demonstrated by this better result.
A unique AES S-box was developed in [9,10], which performed comparable to standard S-boxes and improved security by better adhering to the Strict Avalanche Criterion (SAC), Bit Independence Criterion (BIC), and enhanced algebraic complexity. In order to enhance security for online transactions, banking, and e-commerce, an AES variation in [11] added dynamic S-box and key generation, making ciphertexts more complex and fortifying defenses.
In [12], a real-time keystream technique based on the Chebyshev map was presented, offering strong picture encryption that complied with NIST requirements, but it took a considerable amount of time to implement, which affected speed. With little overhead, an improved AES approach in [13] that used a 256-bit random number generator for a randomized S-box increased security in IoT and military applications. Based on a substitution-permutation network (SPN), the Small Lightweight Cryptographic Algorithm (SLA) in [14] provided faster encryption than Feistel-based ciphers, which made it perfect for embedded systems such as RFID tags and wireless sensor nodes. In order to increase confusion, a permutation phase was added to AES in [15,16] by utilizing chaotic theory. After statistical and NIST assessments, the technique, which used two DNA sequences and a 3D logistic map, finished encryption and decryption in a matter of seconds. Shifting, a circle map, and password keys were used in [17,18] to create a novel S-box generating technique that showed intricate output procedures and dynamic interactions. These encryption methods are compiled in Table 1, which also highlights their advantages and disadvantages. The majority of methods have poor security, excessive complexity, and slow speed. With a lightweight approach that increases process efficiency while preserving strong data security, our research seeks to address these issues [19,20].

III. BACKGROUND AND METHODOLOGY

One popular symmetric encryption scheme for protecting digital data is called Advanced Encryption Standard (AES). The National Institute of Standards and Technology (NIST) created it as a standard in 2001 and is renowned for its dependability, effectiveness, and adaptability to a wide range of uses, including safeguarding financial and governmental data as well as personal data. This is a thorough description of how AES operates Important AES Features:

1. Symmetric Encryption: AES requires the sender and recipient to safely exchange the key because it uses the same key for both encryption and decryption.

2. Block Cipher: AES uses data blocks that are 128 bits (16 bytes) in size.

3. Key Lengths: The three key sizes that AES offers are 128, 192, and 256 bits. As the key length increases, so does the computational complexity and security.

4. Encryption Rounds: AES-128: ten rounds' AES-192: twelve rounds 14 rounds of AES-256 AES Encryption Process: With the exception of the last round, which skips one step, AES encryption consists of multiple processing rounds with four primary phases each. This is how each round works, The first round is Add Round Key, which XORs each block byte with a matching key byte.

2. Main Rounds (Repeated according to the size of the key):
Sub Bytes: A value from an S-box (substitution box) is used to replace each byte in the block. Because of the confusion and non-linearity this creates, the encryption is impervious to both linear and differential cryptanalysis.

Shift Rows: The block's rows are moved to the left using various offsets. In this stage, the bytes are rearranged to introduce diffusion.

Mix Columns: Matrix multiplication over a finite field is used to change each column in the block. This improves diffusion by further dispersing the input data throughout the block.
Add Round Key: Part of the enlarged key is XORed with the current block.

3. Final Round: Add Round Key (without Mix Columns) Sub Bytes Shift Rows Procedure for Decryption,
In order to recover the original plaintext from the ciphertext, AES decryption entails reversing the encryption steps: Inverse Sub Bytes, Inverse Shift Rows, Inverse Mix Columns, and Add Round Key.

Key Expansion: A sequence of round keys is produced for use in every encryption and decryption step by expanding the original key. By guaranteeing that every round has a distinct key, this modification strengthens the algorithm's defense against brute-force attacks.

Applications & Security: When properly implemented, AES is quite secure, particularly when using larger keys like 256 bits. Because there are so many potential keys, it is immune to the majority of known attacks, including brute force.

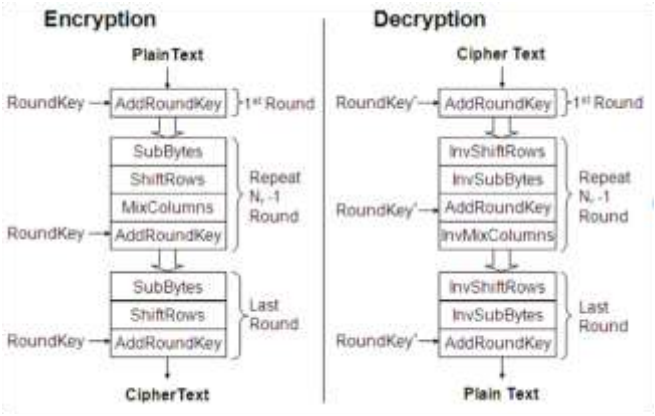

Fig. 1 Encryption and Decryption Flow Chart

Older, less secure encryption algorithms like DES (Data Encryption Standard) have been replaced by AES, which has become a global standard. It is well-known for striking a balance between strong security, performance, and ease of use, which makes it appropriate for a variety of gadgets, from powerful servers to tiny IoT sensors.

1. Creation of Keys To improve security, a brand-new two-step key generating technique is presented:
Step 1: Makes use of a novel chaotic function that is very sensitive to changes in variables, guaranteeing strong encryption and high unpredictability.
Step 2: To increase the generated key's complexity and security, a three-dimensional Lorenzo function is implemented.
For consistency and dependability, the created key is used in all encryption rounds, following the AES encryption scheme.

2.Boxes for Dynamic Substitution To maximize encryption, two dynamic substitution boxes are created:

Even and Odd Rounds: For even and odd rounds, a different substitute box is designated, guaranteeing unpredictable and dynamic changes throughout the encryption process. Performance: By keeping the encryption speed of these substitution boxes to one millisecond per operation, real-time processing capabilities are improved.

3. The Function of Circular Permutation A bit-level circular permutation function is used in place of the conventional AES mix column operation:

Improved Speed and Performance: The goal of this change is to speed up the encryption process without sacrificing the reliability and integrity of data protection.

4. Evaluation and Simulation To determine its capabilities, the suggested method is put through a rigorous simulation and review process:

Metrics of Performance: With an emphasis on its suitability for IoT data, the method is evaluated for encryption speed, randomness, and overall effectiveness. Validation of Security: The suggested method's security strength and performance metrics are verified using the National Institute of Standards and Technology's (NIST) benchmark tests. Hardware specifications. The following hardware elements are necessary in order to execute and assess the suggested algorithm:

IoT Devices: To make sure the technique works on various IoT platforms, real-time encryption and decryption procedures are tested on these devices. Implementing the enhanced AES algorithm to gauge speed and efficiency in hardware contexts requires high-performance FPGA or ASIC platforms. Hardware modules for cryptography are required in order to facilitate the execution of chaotic functions and three-dimensional calculations. Software prerequisites The following software tools are necessary for the algorithm's creation, simulation, and validation.

The method is designed, simulated, and implemented on FPGA platforms using the Xilinx Vivado or ISE Design Suite.
MATLAB or Python: Used to do initial algorithmic simulations and implement the Lorenzo and chaotic functions.
The NIST Test Suite is a collection of software tools used to carry out standardized tests to assess the performance and security of algorithms. Hardware description and cryptographic method implementation on Xilinx FPGA devices require Verilog or VHDL. In order to overcome the drawbacks of traditional approaches and guarantee excellent performance and security, this research attempts to offer a holistic solution by incorporating cutting-edge cryptographic algorithms into a lightweight and effective encryption algorithm designed for IoT contexts.

IV. SIMULATION AND RESULTS

Assessment of Algorithm Performance and Sensitivity This section describes the analysis of key sensitivity, plaintext block dimensions, and comparative performance measures. A Lenovo laptop running Python 3.7 and Windows 10 (64-bit OS) with an Intel Core i5-5200U processor operating at 2.20 GHz and 8 GB of RAM was used for the simulations. Key Creation and Sensitivity of Algorithms Since the parameters of a chaotic map serve as the basis for the key generation process, high sensitivity is ensured: even slight modifications to the input parameters produce notable key differences. This characteristic increases the encryption's strength and unpredictability. The algorithm requires two-byte inputs for creation and depends on the S-box for cryptographic security. The output is significantly altered by changes to these inputs or chaotic map parameters, highlighting the algorithm's sensitivity and strong resilience to cryptanalysis. S-Box Cryptographic Analysis Key cryptographic criteria, including balancedness, nonlinearity, algebraic complexity, the Strict Avalanche Criterion (SAC), and differential cryptanalysis, were used to assess both the even-round and odd-round S-boxes. The outcomes showed that the suggested S-boxes successfully satisfy these requirements. With only one millisecond needed for each operation, the S-box generation and implementation times were extremely efficient. Padding and Block Division The 16-byte blocks of text data were separated, and block

padding was used to make sure the final block had the right size for encryption.

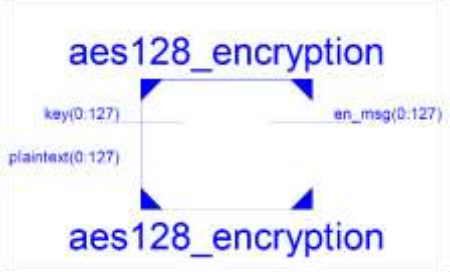

Fig. 2 Encryption RTL 1

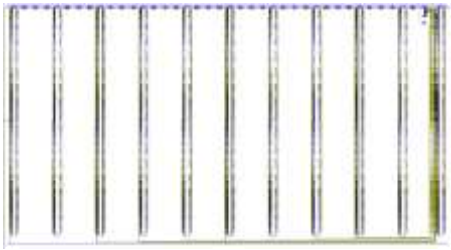

Fig. 3 Encryption RTL 2

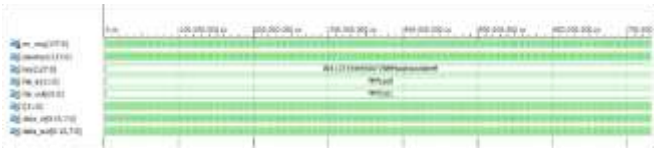

Fig. 4 Encryption SIM

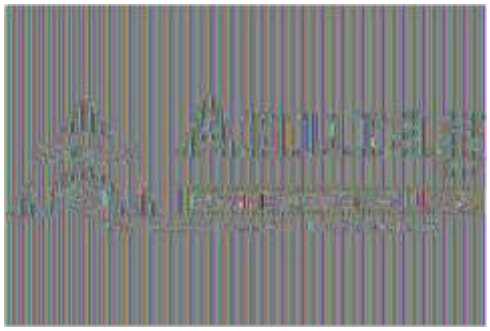

Fig. 5 Encryption Output Anurag

Fig. 6 Decryption RTL 2

Fig. 7 Decryption RTL 2

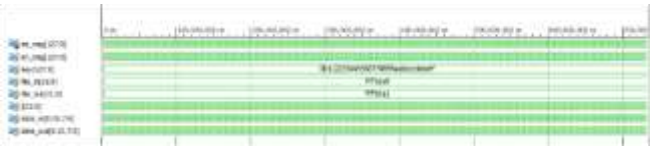

Fig. 8 Decryption RTL 2

Fig. 9 Decryption Output Anurag

The suggested technique proved to be highly effective in various tasks, making it appropriate for encryption in real time. Encryption and Decryption Performance After testing, it was discovered that the encryption and decryption procedures were perfect for sending secure images across wireless communication channels. The visual outcomes of encrypted and decrypted photos are shown in Figure 2, demonstrating the efficacy of the suggested method. Comparison of CPU Time and Performance. The CPU time required for key generation, S-box formation, and encryption/decryption cycles was used to evaluate the suggested algorithm's performance. A comparison with regular AES and other approaches was conducted. Better processing times were shown by the improved AES algorithm, which substituted a circular permutation function for the mix column and included two dynamic S-boxes. Significant time gains are shown by the full encryption and decryption results for plaintext and related ciphertext. Results of the NIST Test
For quality control, the suggested approach was put through the NIST test suite. The algorithm's excellent data handling and security performance was validated by this rigorous assessment. The results of several NIST tests are shown the suggested encryption method's resilience and dependability. The efficiency of the created approach is demonstrated by the time comparison (KB/s) with other algorithms.

V. CONCLUSION

With an emphasis on lightweight performance to accommodate their quick and changing nature, this work improves the AES algorithm for Internet of Things devices. Achieving high security with generation times under 1.2 milliseconds, important enhancements include the use of dynamic S-boxes for even and odd rounds and dual key generation utilizing a novel chaotic map and a 3D Lorenzo map. The mix column function is replaced with the bit-level cyclic permutation, which increases handling speed and versatility for color and grayscale images. NIST testing ensured safe handling of IoT data by verifying adherence to encryption standards. The algorithm changes and revised key generation technique satisfy encryption standards while adjusting to changing cybersecurity demands. These

techniques will be expanded to video data encryption in future research, and equally effective cryptographic algorithms for the Internet of Things will be created.

REFERENCES

[1] Jassim, S. A., & Farhan, A. K. (2022). Designing a New Lightweight AES Algorithm to Improve the Security of the IoT Environment. *IRAQI JOURNAL OF COMPUTERS, COMMUNICATIONS, CONTROL AND SYSTEMS ENGINEERING*, *22*(2), 96-108.

[2] Fadhil, M. S., Farhan, A. K., & Fadhil, M. N. (2021). A lightweight aes algorithm implementation for secure iot environment. *Iraqi Journal of Science*, 2759-2770.

[3] Gupta, A., & Jaiswal, M. (2017, May). An enhanced AES algorithm using cascading method on 400 bits key size used in enhancing the safety of next generation internet of things (IOT). In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 422-427). IEEE.

[4] Farooq, U., Mushtaq, M., & Bhatti, M. K. (2020, October). Efficient AES implementation for better resource usage and performance of IoTs. In *CYBER 2020-5th International Conference on Cyber-Technologies and Cyber-Systems*.

[5] Al_Azzawi, R. M. A., & Al-Dabbagh, S. S. M. (2023). A lightweight encryption algorithm to secure iot devices. *MINAR Int. J. Appl. Sci. Technol*, *5*(03), 37-62.

[6] Arman, S., Rehnuma, T., & Rahman, M. (2020, December). Design and implementation of a modified AES cryptography with fast key generation technique. In *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)* (pp. 191-195). IEEE.

[7] Abd Zaid, M., & Hassan, S. (2019). Modification advanced encryption standard for design lightweight algorithms. *J. Kufa Math. Comput.*, *6*(1), 21-27.

[8] Babu, T., Murthy, K. V. V. S., & Sunil, G. (2011). Aes algorithm implementation using arm processor. In *2nd International Conference and workshop on Emerging Trends in Technology (ICWET) Proceedings published by International Journal of Computer Applications (IJCA)*.

[9] Roy, S., Stavrou, A., Mark, B. L., Zeng, K., PD, S. M., & Khasawneh, K. N. (2022, April). Characterization of AES Implementations on Microprocessor-based IoT Devices. In *2022 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 55-60). IEEE.

[10] Prasad, B. D. C. N., & Prasad, P. K. (2010). A Performance Study on AES algorithms. *International Journal of computer science and information security*, *8*(6), 128-132.

[11] More, S., & Bansode, R. (2015). Implementation of AES with time complexity measurement for various input. *Global Journal of Computer Science and Technology: E Network, Web & Security*, *15*(4), 11-20.

[12] Alassaf, N., & Gutub, A. (2019). Simulating light-weight-cryptography implementation for IoT healthcare data security applications. *International Journal of E-Health and Medical Communications (IJEHMC)*, *10*(4), 1-15.

[13] Mamvong, J. N., Goteng, G. L., Zhou, B., & Gao, Y. (2020). Efficient security algorithm for power-constrained IoT devices. *IEEE Internet of Things Journal*, *8*(7), 5498-5509.

[14] Hazzaa, F., Hasan, M. M., Qashou, A., & Yousef, S. (2024). A New Lightweight Cryptosystem for IoT in Smart City Environments. *Mesopotamian Journal of CyberSecurity*, *4*(3), 46-58.

[15] Dang, T. N., & Vo, H. M. (2019, February). Advanced AES algorithm using dynamic key in the internet of things system. In *2019 IEEE 4th international conference on computer and communication systems (ICCCS)* (pp. 682-686). IEEE.

[16] Jat, D. S., & Gill, I. S. (2020, November). Enhanced advanced encryption standard with randomised s box. In *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)* (pp. 1-6). IEEE.

[17] Bharathi, R., & Parvatham, N. (2020). LEA-SIoT: hardware architecture of lightweight encryption algorithm for secure IoT on FPGA platform. *International Journal of Advanced Computer Science and Applications*, *11*(1).

[18] Mohammad, H. M., & Abdullah, A. A. (2022). Enhancement process of AES: a lightweight cryptography algorithm-AES for constrained devices. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *20*(3), 551-560.

[19] AbuRass, S., & Qatawneh, M. (2018). Performance Evaluation of AES algorithm on Supercomputer IMAN1. *International Journal of Computer Applications*, *179*(48), 32-34.

[20] Shah, A., Shah, S., Patel, H., & Shah, N. (2023). LSA: A LIGHTWEIGHT SYMMETRIC ENCRYPTION ALGORITHM FOR RESOURCE-CONSTRAINED IOT SYSTEMS. *Reliability: Theory & Applications*, *18*(3 (74)), 44-58.