Secure Real-Time Healthcare Monitoring in Medical IOT Using Modified Encryption Algorithm

Sowmiya M
Department of ECE
PSG Institute of Technology and Applied Research
Tamil Nadu, India
sowmiya@psgitech.ac.in

Sasmitha U
Department of ECE
PSG Institute of Technology and Applied Research
Tamil Nadu, India
sasmithauthamaraj@gmail.com

Janarthanan K
Department of ECE
PSG Institute of Technology and Applied Research
Tamil Nadu, India
janak9786800@gmail.com

Mithilesh Ravi Shankaran
Department of ECE
PSG Institute of Technology and Applied Research
Tamil Nadu, India
drsmithilesh@gmail.com

line 1: 5th Given Name Surname
Department of ECE
PSG Institute of Technology and Applied Research
Tamil Nadu, India
rameshragavendran1@gmail.com

2025 International Conference on Next Generation Computing Systems (ICNGCS) | 979-8-3315-9116-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICNGCS64900.2025.11182773

***Abstract*—In the rapidly evolving field of the Medical Internet of Things (MIoT), ensuring the confidentiality and integrity of sensitive health data is critical. This paper presents a Modified Advanced Encryption Standard (AES) algorithm specifically designed for secure, real-time healthcare monitoring applications. The proposed enhancements include four major contributions: (1) dynamic S-box generation using Fisher-Yates shuffle seeded with device-specific values to increase unpredictability; (2) a hybrid Counter-Cipher Block Chaining (CTR-CBC) mode that combines high throughput with strong diffusion; and (3) lightweight key expansion using SHA-256, significantly reducing memory and computational overhead and (4) Cipher-based Message Authentication Code (CMAC) to provide integrity. Comparative analysis with standard AES demonstrates improved encryption speed, better adaptability to embedded environments, and increased resilience to cryptanalytic attacks, making it highly suitable for MIoT systems.**

Keywords— Modified AES, MIoT, Dynamic S-Box, SHA-256, Hybrid CTR-CBC Mode, CMAC, Real-Time Encryption.

I. Introduction

In recent years, the integration of healthcare with advanced communication technologies has given rise to the Medical Internet of Things (MIoT), revolutionizing patient care through smart, connected devices such as wearable ECG monitors, glucose sensors, and pulse oximeters. These devices continuously collect and transmit critical health data in real time to centralized platforms for analysis and decision-making, enabling timely interventions, personalized treatment plans, and reduced strain on healthcare systems. MIoT enhances the accessibility and scalability of healthcare, particularly in remote or non-clinical settings.

However, this transformation introduces significant challenges—most notably data security, privacy protection, and system efficiency. MIoT devices typically operate under tight resource constraints such as limited power, memory, and processing capabilities, making traditional encryption methods like AES less practical. Although AES is secure, its computational demands can lead to latency and performance issues in real-time medical environments. This project addresses these concerns by proposing lightweight, adaptable cryptographic techniques specifically designed for MIoT systems to ensure data confidentiality and integrity without compromising efficiency, scalability, or compliance with healthcare standards.

II. Literature Survey

Lin et al. introduced a high-performance AES-GCM architecture optimized with parallel and pipelined designs on FPGA, achieving 57.17 Gbps throughput and efficient BRAM-based S-box implementation. [1] Nagaraju et al. enhanced AES by reducing computational overhead using biometric-based dynamic key generation, improving energy and area efficiency. [2] Adeniyi et al. proposed a modified AES for cloud-based medical data with a faster final encryption round, achieving reduced encryption time and stronger avalanche effect. [3] Assa-Agyei et al. compared AES, Blowfish, and TwoFish, identifying AES as fastest in encryption, but highlighted the need for optimization for practical use. [4] Rehman et al. presented a hybrid AES-ECC model to strengthen cloud storage security, combining AES speed with ECC's authentication for reduced breach risk. [5] Maria Jan et al. combined AES, RC4, and DES for faster and memory-efficient cloud encryption, adaptable by data type and suitable for real-time applications. [6] Jacinto et al. integrated AES-CBC with LSB steganography for triple-layered data protection in images, enhancing imperceptibility and execution speed. [7] Aditya et al. optimized AES power consumption using SSTL IO standards on Spartan-7 FPGA, emphasizing dynamic power influence on total consumption. [8] Sridevi et al. proposed an area-efficient, pipelined AES implementation on FPGA, suitable for real-time applications, with future plans for added cryptographic features. [9] Gebeyehu et al. replaced AES's MixColumns with bitwise reverse transposition in EE-AES, improving speed and throughput for low-power devices. [10] Kim et al. optimized AES-CTR and AES-GCM on 8-bit microcontrollers, using custom LUTs and efficient binary multipliers to resist side-channel attacks. [11][12]

III. Existing AES Algorithm

Before The Advanced Encryption Standard (AES), developed by NIST in 2001, is a symmetric block cipher and the global standard for data security across sectors like finance, healthcare, and embedded systems. It operates on 128-bit data blocks with key sizes of 128, 192, or 256 bits, undergoing 10, 12, or 14 transformation rounds respectively. Each round includes SubBytes (non-linear byte substitution using an S-box), ShiftRows (row-wise permutation), MixColumns (Galois Field matrix multiplication), and AddRoundKey (XOR with expanded round key). These

979-8-3315-9116-8/25/$31.00 ©2025 IEEE

TABLE V. COMPARISON OF STANDARD AES AND MODIFIED AES-CMAC

Parameters	Standard AES			Modified AES		
Key Size	***128 bits***	***192 bits***	***256 bits***	***128 bits***	***192 bits***	***256 bits***
Speed for Encryption	182.716 KB/s	148.433 KB/s	127.963 KB/s	162.799 KB/s	134.344 KB/s	117.162 KB/s
Speed Decryption	145.176 KB/s	117.332 KB/s	101.467 KB/s	155.682 KB/s	129.237 KB/s	112.963 KB/s
Time for Encryption	1.010 ms	1.428 ms	1.442 ms	1.113 ms	1.373 ms	1.575 ms
Time for Decryption	1.271 ms	1.573 ms	1.818 ms	1.185 ms	1.428 ms	1.633 ms
Bytes Transferred	553527	553527	553527	553527	553527	553527

From Table.5 it clearly depicts a detailed performance analysis of the Standard AES and the proposed Modified AES encryption algorithm across three different key sizes: 128-bit, 192-bit, and 256-bit. The comparison was based on four key metrics: encryption speed, decryption speed, encryption time, and decryption time, using a fixed data size of 553,527 bytes.

D. Real-Time Implementation

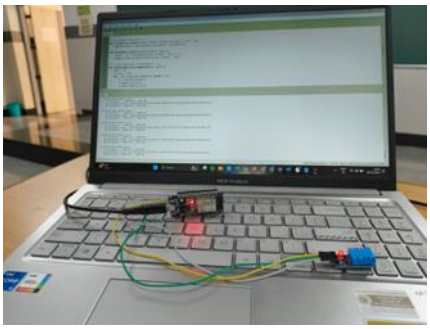

Fig. 18. Real-Time Implementaion

Figure 18 showcases the real-time implementation and evaluation of standard and modified AES algorithms on an ESP32 using a DHT11 sensor. The setup captures temperature and humidity data, processes it via Micro Python, and encrypts it using both AES variants. While both achieve correct decryption, the Modified AES offers reduced output size (18 bytes vs. 32 bytes) without compromising accuracy or processing time (~7 ms), indicating improved efficiency. This validates the proposed algorithm's suitability for secure, bandwidth-efficient real-time IoT applications on resource-constrained devices.

E. Security Evaluation and Empirical Cryptoanalysis

To ensure that the proposed Modified AES design is not only performant but also secure, an extensive Python-based evaluation framework was developed to simulate various cryptographic tests. Beyond benchmarking speed the script rigorously evaluates cryptographic strength through multiple test suites.

1) Known-Plaintext Pattern Leakage: The standard AES exhibits visible ciphertext repetition for repetitive inputs (e.g., b"A"*32), clearly showing its inability to obscure data patterns. In contrast, the proposed Hybrid AES masks such patterns effectively using chaining and key-based S-box variation, ensuring ciphertext blocks are indistinguishable even with repeated inputs.

2) Avalanche Effect Analysis: The framework tests how a 1-bit flip in either the plaintext or key impacts the ciphertext. It achieved ~50% avalanche on average—demonstrating high diffusion and robustness.

3) Differential Cryptanalysis Simulation: Thousands of plaintext pairs differing by a fixed XOR pattern were encrypted to analyze the distribution of output differences. The final design showed no dominant output differentials, indicating the absence of exploitable statistical biases.

This security-focused validation complements performance metrics and demonstrates that the proposed algorithm is not only lightweight and efficient but also cryptographically sound. The testing framework, written entirely in Python, played a pivotal role in iteratively refining and verifying the cipher's design for Medical IoT use.

VI. CONCLUSION

The proposed hybrid AES encryption scheme offers a robust solution tailored to the stringent requirements of Medical IoT systems, where real-time data transmission and low-power operation are crucial. By integrating SHA-256-based key expansion, dynamic S-box generation, and a CTR-CBC hybrid mode, the system effectively balances security, performance, and resource efficiency. The inclusion of CMAC ensures strong data integrity and authentication, which is vital for protecting sensitive healthcare information. Experimental results show that the modified AES maintains competitive encryption speeds while enhancing decryption performance and consistency across various key sizes—beneficial for embedded medical devices that demand predictable timing and energy efficiency. Additionally, the use of session-specific keying and dynamic S-boxes boosts entropy, strengthening resistance to cryptanalytic attacks without burdening device resources. With its modular architecture, this framework lays a solid foundation for future improvements, including quantum-resistant primitives, context-aware encryption, and energy optimization through hardware acceleration—positioning it as a scalable and secure solution for both current and emerging healthcare technologies

REFERENCES

[1] Lin, M. B., & Chuang, J. H. (2023), "The design of a high-throughput hardware architecture" for the AES-GCM algorithm. IEEE Transactions on Consumer Electronics, 70(1), 425-432.

[2] Nagaraju, S., Nagendra, R., Balasundaram, S., & Kumar, R. K. (2023). "Biometric key generation and multi round AES crypto system for improved security". Measurement: Sensors, 30, 100931.

[3] Adeniyi, A. E., Abiodun, K. M., Awotunde, J. B., Olagunju, M., Ojo, O. S., & Edet, N. P. (2023). "Implementation of a block cipher algorithm for medical information security on cloud environment: using modified advanced encryption standard approach". Multimedia Tools and Applications, 82(13), 20537-20551.

[4] Asif, A., Charters, P. F., Thompson, C. A., Komber, H. M., Hudson, B. J., & Rodrigues, J. C. L. (2022). "Artificial intelligence can detect left ventricular dilatation on contrast-enhanced thoracic computed tomography relative to cardiac magnetic resonance imaging". The British Journal of Radiology, 95(1138), 20210852.

[5] Rehman, S., Talat Bajwa, N., Shah, M. A., Aseeri, A. O., & Anjum, A. (2021). "Hybrid AES-ECC model for the security of data over cloud storage". Electronics, 10(21), 2673.

[6] Jan, M., Shahzad, Q., & Afsar, S. (2022). "Securing the Cloud Storage by Using Different Algorithms of Cryptography". Int. J. Sci. Res. in Computer Science and Engineering Vol, 10(2).

[7] Edwar, J. G., & Holman, M. A. (2022). "Enhanced Security: Implementation of Hybrid Image Steganography Technique using Low-Contrast LSB and AES-CBC Cryptography". International Journal of Advanced Computer Science and Applications, 13(8).

[8] Aditya, Y., & Kumar, K. (2022). "Implementation of novel power efficient AES design on high performance FPGA". NeuroQuantology, 20(10), 5815.

[9] Priya, S. S. S., Karthigaikumar, P., & Teja, N. R. (2022). "FPGA implementation of AES algorithm for high-speed applications". Analog integrated circuits and signal processing, 1-11.

[10] Zinabu, N. G., & Adere, K. (2022). "Enhanced Image Cipher and Decipher Speed of Advanced Encryption Standard Algorithms for Embedded Devices". Available at SSRN 4765737.

[11] Kim, K., Choi, S., Kwon, H., Kim, H., Liu, Z., & Seo, H. (2020)."PAGE—practical AES-GCM encryption for low-end Microcontrollers". Applied Sciences, 10(9), 3131.

[12] Babiuch, M., Foltýnek, P., & Smutný, P. (2019, May). Using the ESP32 microcontroller for data processing. In *2019 20th International Carpathian Control Conference (ICCC)* (pp. 1-6). IEEE.

[13] Pravalika, V., & Prasad, C. R. (2019). Internet of things based home monitoring and device control using Esp32. International Journal of Recent Technology and Engineering, 8(1S4), 58-62.

[14] Muruganantham Sowmiya, B. Banu Rekha, Elangeeran Malar and K.R. Ashwin Kumaran (2022)."Hierarchical learning model for early prediction of coronary artery atherosclerosis" Available at International Journal of Operational Research**.**

[15] M. Sowmiya, B. Banu Rekha and E. Malar. (2025), "Optimized heart disease prediction model using a meta-heuristic feature selection with improved binary salp swarm algorithm and stacking classifier". Computers in Biology and Medicine, Volume 191, 110171.