ISSN: 2393-8048
International Advance Journal of Engineering, Science and Management (IAJESM)
Multidisciplinary, Multilingual, Indexed, Double Blind, Open Access, Peer-Reviewed, Refereed-International Journal.
SJIF Impact Factor =8.152, July-December 2025, Submitted in September 2025

# Lightweight Cryptographic Algorithm for Enhanced Security in IOT Based Smart Diabetic Monitoring System

Ms. Urvija Raina, Research Scholar, Nirwan University, Jaipur
Prof. (Dr.) Amit Singla, Professor, Nirwan University, Jaipur

## Abstract

In the rapidly evolving era of the Internet of Things (IoT), data security and computational efficiency have become crucial factors, especially within healthcare applications where continuous monitoring and wireless data transmission are routine. Traditional cryptographic techniques such as AES and RSA, while robust, impose significant computational and energy burdens on resource-limited IoT devices. The objective of this paper is to design real-time efficient method for providing data protection in IoT-based smart diabetic monitoring system. The proposed Enhanced Lightweight Cryptographic Algorithm (ELCA) employs an optimized ARX (Addition–Rotation–XOR) structure that supports fast operations, strong diffusion and strong resistance to timing and side-channel attacks. The model further incorporates advanced key management and nonce-tracking mechanisms to ensure non-repetition, data integrity and end-to-end authentication. Experimental results revealed that the proposed algorithm exhibited superior scalability and stable performance across different data sizes, confirming its suitability for resource-constrained IoT environments.

**Keywords: Internet of Things(IoT), Lightweight cryptography, Authenticated encryption, Resource-constrained devices, Key management, Nonce tracking**

## 1. INTRODUCTION:

The ability to process, analyze, and interpret massive amounts of data has revolutionized every sphere of modern life. Internet of Things (IoT) stands out as one of the most significant innovations of the 21st century digital revolution. It extends the boundaries of traditional computing by connecting billions of devices over the internet, enabling them to collect, share, and analyze data autonomously[1]. The smart ecosystem has found immense application in various domains offering enhanced convenience, efficiency, and safety. By combining IoT with modern technologies such as cloud computing and artificial intelligence (AI), these systems can perform real-time monitoring, predictive analysis, and intelligent decision-making[2].

Internet of Medical Things (IoMT) connects medical devices, sensors, and healthcare applications into an integrated network that enables continuous monitoring of patients' vital parameters, remote diagnostics, and personalized medical care. Such systems are particularly beneficial for managing chronic diseases like diabetes where real-time data collection and alert mechanisms can prevent emergencies and improve patient outcomes. However, the growing interconnectivity and data dependency of IoMT systems bring along severe security and privacy challenges as sensitive health data is vulnerable to interception, tampering, and unauthorized access[3].Traditional cryptographic mechanisms, though strong in conventional computing environments, often fall short when applied to IoT-based healthcare systems due to latency,high computational complexity and energy consumption[4]. This necessitates development of lightweight cryptographic algorithms that provide robust data protection without compromising efficiency. The goal is to strike an optimal balance between security, performance, and scalability.

This study aims to design and implement an enhanced lightweight cryptographic algorithm tailored for secure communication in IoT-based diabetic monitoring systems. The proposed approach focuses on protecting sensitive health information collected from sensors, ensuring confidentiality, integrity, and authenticity of transmitted data by combining the efficiency of stream ciphers with optimized key management and authentication protocols.

## 2. LITERATURE REVIEW:

**Elminaam et al. (2008)** and **Haque et al. (2018)** conducted performance evaluations of traditional symmetric algorithms like AES, Blowfish, and RC4, concluding that although such algorithms guarantee strong confidentiality, their processing overhead and energy demands make them impractical for small, battery-operated devices. This limitation has driven the development of lightweight algorithms optimized for reduced latency and power use.

**Valsalan et al. (2020)** demonstrated secure IoT-enabled health monitoring systems that rely on sensor data transmission and cloud-based analytics, underscoring the importance of efficient encryption mechanisms to protect medical information.

**Mohammed et al. (2023)** uses a health tracking system that leverages a GSM/GPRS/GNSS HAT module to obtain patient location when needed. Sensor data are transmitted to cloud storage and persisted in a MySQL database, while a cross-platform mobile application provides doctors and patients with live access to readings for synchronous monitoring and rapid clinical decision-making.

**Jadhav (2019)** provide systematic comparisons of existing LWC algorithms which led to the fact that no single lightweight cipher provides an optimal balance among cost, speed, and security, reinforcing the need for adaptive or hybrid cryptographic models that can dynamically adjust to IoT device constraints.

**Al-Husainy et al. (2021)** further explored modifications of existing ciphers such as XXTEA and DNA-based encryption, achieving enhanced randomness, diffusion, and avalanche effects critical for IoT data confidentiality.

**Mousavi et al. (2021)** emphasized the criticality of integrating efficient encryption with authentication protocols for protecting sensitive health data transmitted through smart systems. The study demonstrates that combining streamlined key management with lightweight symmetric encryption can significantly mitigate threats like data tampering and unauthorized access.

Across these studies, a consistent theme emerges: while lightweight block ciphers have dominated early IoT research, stream ciphers are increasingly recognized for their suitability in real-time, continuous data environments. Stream ciphers operate on smaller data units, require fewer computational cycles, and allow for faster encryption and decryption processes, making them ideal for IoT devices with limited power and memory. However, existing lightweight stream cipher designs often struggle to balance efficiency with robust key scheduling and authentication features. Addressing these challenges will play a pivotal role in realizing the full potential of IoT technology within the healthcare sector and other mission-critical domains.

## 3. Architecture of IoT supported Healthcare System

The study focuses on developing an enhanced lightweight cryptographic mechanism specifically designed for secure and efficient data transmission in IoT-based diabetic monitoring systems while minimizing computational metrics. The architectural design of IoT-enabled healthcare system comprises of three major layers. At the acquisition layer, patient physiological data in form of blood glucose levels, are continuously collected through biomedical sensors embedded in wearable or implantable IoT devices. These sensors are connected to a microcontroller that initiates local real-time encryption using the ELCA algorithm before transmitting the data to ensure protection of sensitive health information. The communication layer handles the secure transfer of encrypted packets to cloud servers through wireless communication protocols. Finally, the cloud layer manages data storage and analytics where decrypted and verified information supports medical monitoring and diagnostic insights.

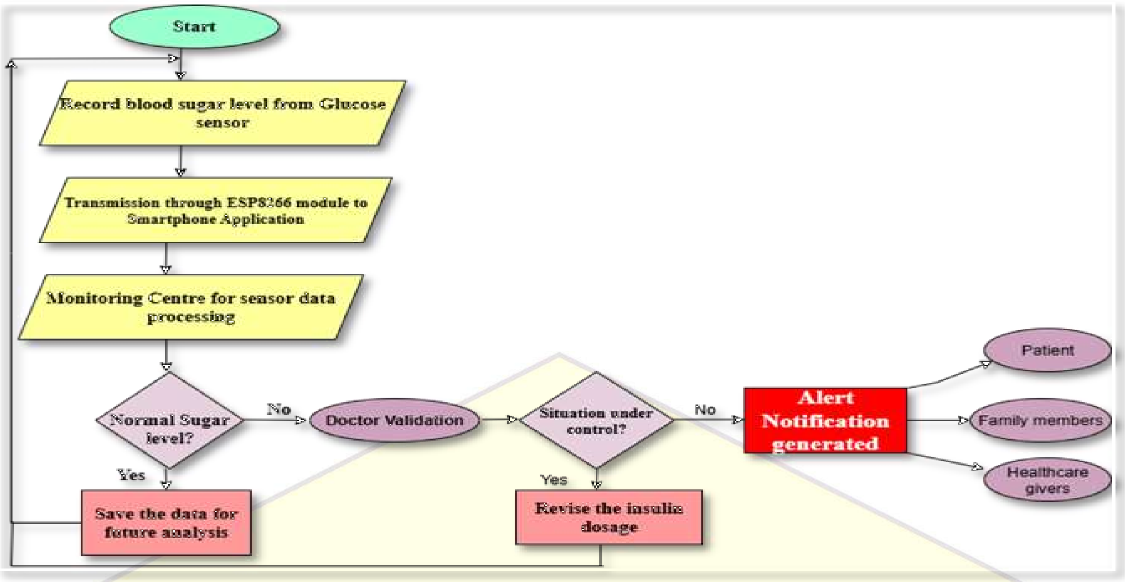
Start
Record blood sugar level from Glucose sensor
Transmission through ESP8266 module to Smartphone Application
Monitoring Centre for sensor data processing
Normal Sugar level?
No
Yes
Save the data for future analysis
Doctor Validation
Situation under control?
No
Yes
Revise the insulin dosage
Alert Notification generated
Patient
Family members
Healthcare givers

**Figure 1 Flowchart of smart diabetic monitoring system**

Figure 1 shows the step by step data flow involved in the functionality of ELCA Algorithm Glucose sensors collect data of patient for the smart diabetic monitoring system and the information is then transmitted via a Wi-Fi network to the mobile application. Data is sent to monitoring center after encryption performed by proposed ELCA algorithm. The data center checks for any abnormally low or high blood sugar level. If everything is normal, the current data is saved for any future analysis. In case of medical emergency, having obtained the processed results, the monitoring center communicates with the doctor (via the doctor's phone). The doctor after reviewing the case in hand decides whether revising insulin dosage can control the situation. The same is communicated to patient on his phone as urgent medical notification and the current data is saved by system automatically. If case requires hospitalization, then an alert notification regarding the same is issued to patient, immediate family members and health care providers.

## 4. Proposed Cryptographic Design Structure:-

The proposed implementation begins with the selection of an appropriate cryptographic strategy. Considering the inherent limitations of IoT devices and the continuous nature of health data transmission, symmetric key cryptography was identified as the most suitable option. Symmetric stream ciphers offer faster execution, lower energy usage, and reduced complexity compared to asymmetric approaches, which are computationally heavy due to their large key sizes. The proposed cryptographic framework employs modified form of ChaCha20 which is a lightweight symmetric cipher as its core encryption mechanism. ELCA follows an ARX (Addition–Rotation–XOR) design as it ensures non-linearity, high diffusion, and resistance against timing and cache-based side-channel attacks[5]. The proposed algorithm is therefore typically optimized for stream-based data, one that ensures fast execution and low memory utilization. It uses a 256-bit key and a 96-bit nonce, generating a keystream that is XORed with plaintext to produce ciphertext. Apart from the 20 round iterations of quarter round functions and mixing operations, the following modifications have been proposed in the ELCA cipher.

For every fixed secret key that is generated, ELCA reuses a precomputed state matrix for its operations. This implies that ELCA reuses the key schedule / initial state by building the initial 16-word state for a given key once. The same is stored for future use. The counter or nonce is only updated per message. For each encryption process, the counter and nonce is only copied or modified before running the quarter-rounds. This saves the cost of reinitializing constants and key words on every block for repeated messages with the same key. However it must be ensured that the counter and nonce words are set correctly and are never reused with the same key. Additionally, the counter needs to be incremented if persistent counters are being used. This approach guarantees fast, constant-time operations suitable for real-time medical data encryption. After transforming the sensor data into ciphertext by the above encryption process, it is then appended with a cryptographic hash or message authentication code (MAC) to ensure data integrity and authenticity. The following components add security features to the ELCA cipher:-

**1. Secure Key Storage & Management:** This feature creates and keeps a secret key safe through the following modules.

**generate_key():** Creates a secret random 256 bit key that is used to lock and unlock the data.

**encrypt_key():** Takes the secret key which is encrypted using AES-GCM, a robust encryption algorithm derived from a user-provided password using PBKDF2HMAC. Some extra random bits (salt and nonce) are also created that are needed for the encryption process. The encrypted key, along with the necessary salt and nonce for decryption, is saved to a file. This means the raw secret key is never stored directly on the device.

**PBKDF2HMAC():** This function adds a layer of security by using the user's password to derive the actual key used for encrypting the main secret key. The high iteration count (100,000) of this key derivation function makes brute-force attacks on the password very difficult. This makes it computationally expensive for an attacker to guess the password and thus the key. decrypt_key(): Takes the scrambled key, those extra random bits, and your password to unlock the secret key.

**save_key() and load_key():** These functions handle saving the scrambled key and the extra bits to a file and loading them back when needed.

**2. Nonce Management:** This part ensures that each time user encrypts data, a unique "nonce" (a number used once) is utilized. This crucial component for security is achieved through Nonce Manager which keeps track of a counter. Reusing a nonce with the same key can compromise the encryption. The nonce is managed persistently to guarantee uniqueness across sessions. Every time there is need to encrypt something, it uses the current count to create a unique nonce and then increases the count for the next time. This count is saved to a file so it remembers where it left off even if the program stops and restarts.

**3. ELCA Cipher:** This is the core of the data protection which uses a fast and secure encrypt() function. The normal plaintext data (like the blood sugar reading), the unique nonce, and the secret key undergo series of ARX transformations also known as Quarter Round Functions to turn the data into an unreadable scrambled version (ciphertext). It also adds a special authentication tag to the ciphertext just to make sure the data hasn't been tampered with.

The decrypt() function takes the ciphertext, the same unique nonce, and the secret key to turn it back into the original readable data. It also checks that the data hasn't been changed since it was encrypted.This combination provides authenticated encryption (AEAD) capabilities to the remote diabetic monitoring system. This dual-layer security model ensures that both encryption and authentication are performed simultaneously, preventing tampering and unauthorized data modification. The authentication mechanism integrated into the ELCA model ensures that only legitimate devices and servers can decrypt and access patient information, strengthening end-to-end trust within the system.

**5. Implementation of Proposed Algorithm:-**

The implementation environment includes simulation and testing of the proposed ELCA model using Python-based cryptographic libraries in Google Colab environment. Real-world diabetic monitoring sensors are emulated so as to transmit health data to a centralized server. The performance of ELCA algorithm is analyzed and compared against standard stream and block ciphers like AES-128 GCM, conventional ChaCha20 cipher, AES-CBC and Triple DES algorithms. Key performance indicators — including encryption and decryption time, throughput, latency, memory usage, and energy consumption — are measured to evaluate the system's efficiency and suitability for IoT applications. The purpose of this analysis is to validate ELCA's suitability for IoT-based healthcare systems, especially in scenarios involving real-time data transmission and processing under resource-constrained conditions. The performance of all algorithms was evaluated across multiple parameters, namely encryption/decryption time, throughput, latency and power consumption. Each parameter was analyzed based on experimental results obtained through simulated executions on varying data sizes (64, 128, 512, 1024, and 10240 bytes) in a Python-based Google Colab environment.

## 6. Results and Discussion:-

The experimental evaluation of the proposed ELCA model has been carried out to assess its performance and efficiency in comparison with other standard lightweight algorithms. The results have been discussed below:

### 6.1 Execution Time (Encryption and Decryption)

Execution time represents total processing duration for encrypting and decrypting data. ELCA consistently recorded the lowest execution times for both operations across all tested data sizes. For 64 bytes input, ELCA required only 0.000005 seconds for encryption and 0.000004 seconds for decryption, outperforming both ChaCha20 and AES-GCM. For larger data packets (10240 bytes), ELCA maintained superior efficiency, completing encryption in 0.000012 seconds—almost 1.5 times faster than AES-GCM.

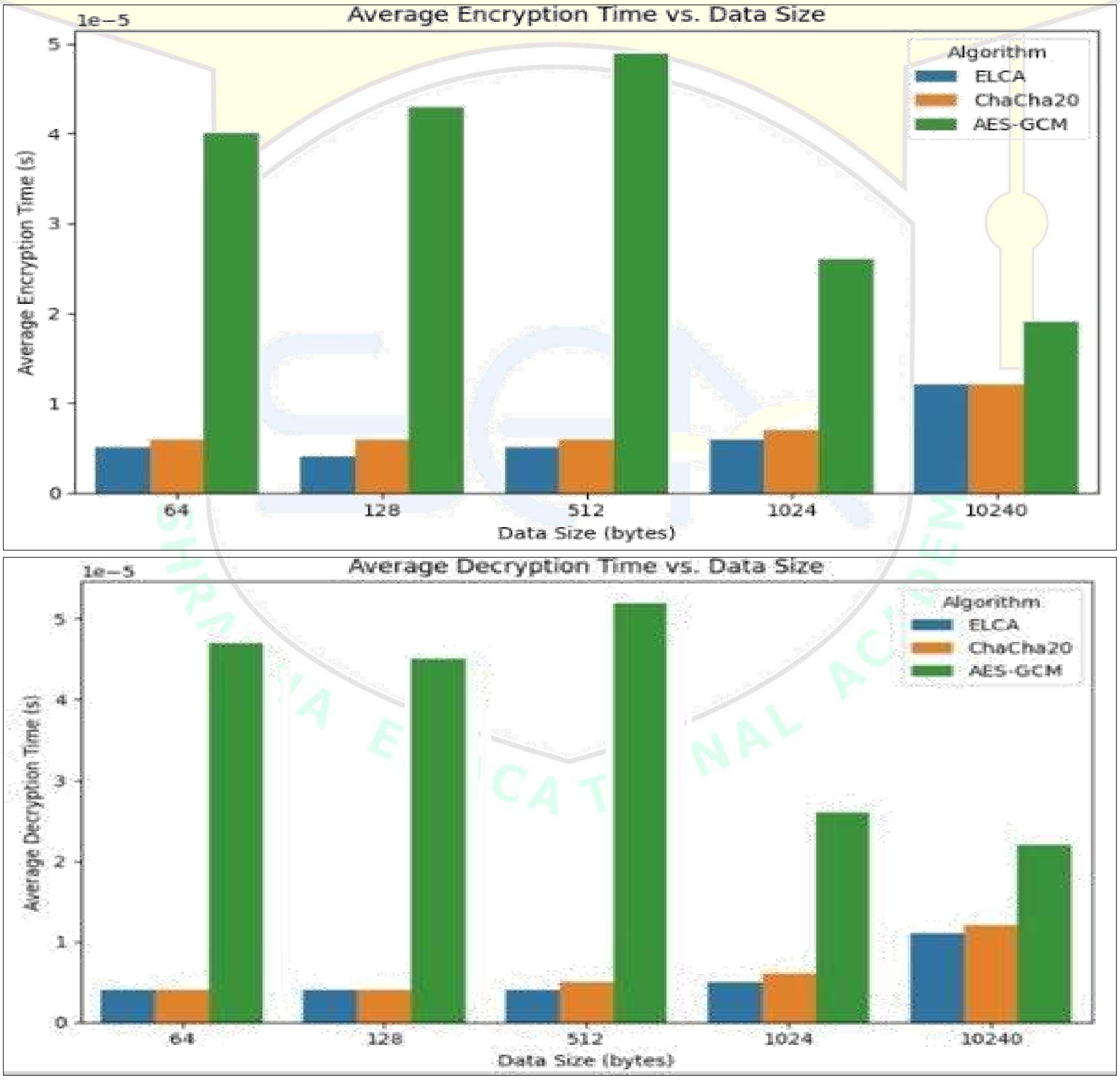

**Fig 2&3 Comparison of avg encryption/decryption time of ELCA, AES-GCM & Chacha20**

When compared with AES-CBC and Triple DES-CBC, ELCA again exhibited remarkable speed, showing average encryption time of 0.000013 seconds for small data sizes and 0.000051 seconds for larger inputs.

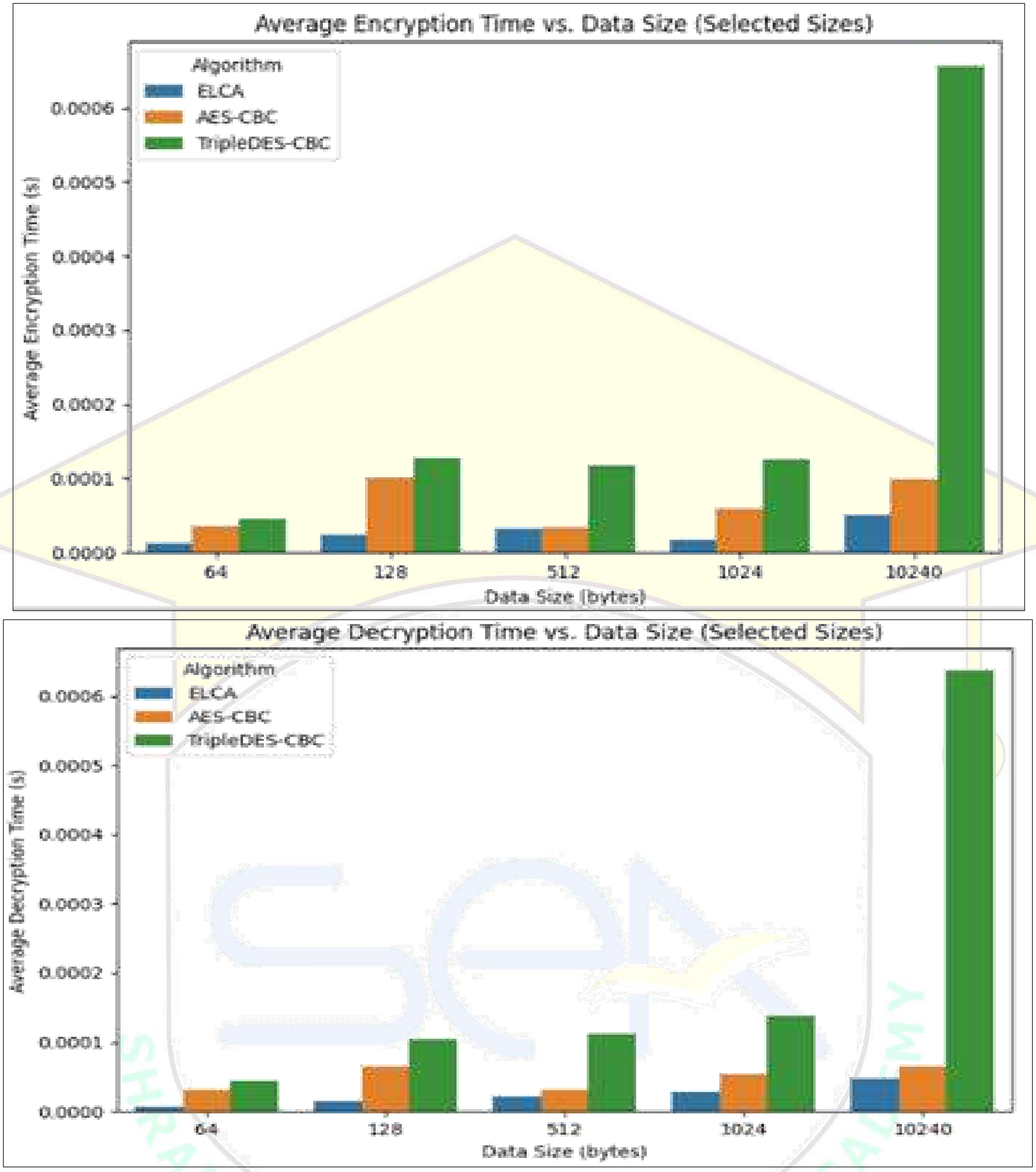

**Fig 4&5 Comparison of avg encryption/decryption time of ELCA, AES-CBC & TripleDES-CBC**

### 6.2 Throughput

Throughput measures amount of data processed per unit time and directly reflects algorithmic efficiency. ELCA achieved the highest throughput values among all tested algorithms, indicating its ability to handle high data volumes efficiently. For 64-byte data, ELCA achieved an encryption throughput of 103.94 Mbps, surpassing ChaCha20 (84.67 Mbps) and AES-GCM (12.64 Mbps). Its decryption throughput also led with 111.73 Mbps, confirming ELCA's superior scalability. As data size increased, ELCA continued to maintain higher throughput showing excellent adaptability without performance degradation. Even when compared to AES-CBC and Triple DES-CBC, ELCA demonstrated far greater throughput (1622.05 Mbps for large data blocks versus AES-CBC's 825.64 Mbps and Triple DES's 124.8 Mbps). These results confirm that the ELCA model achieves a strong balance between processing speed and secure data handling.

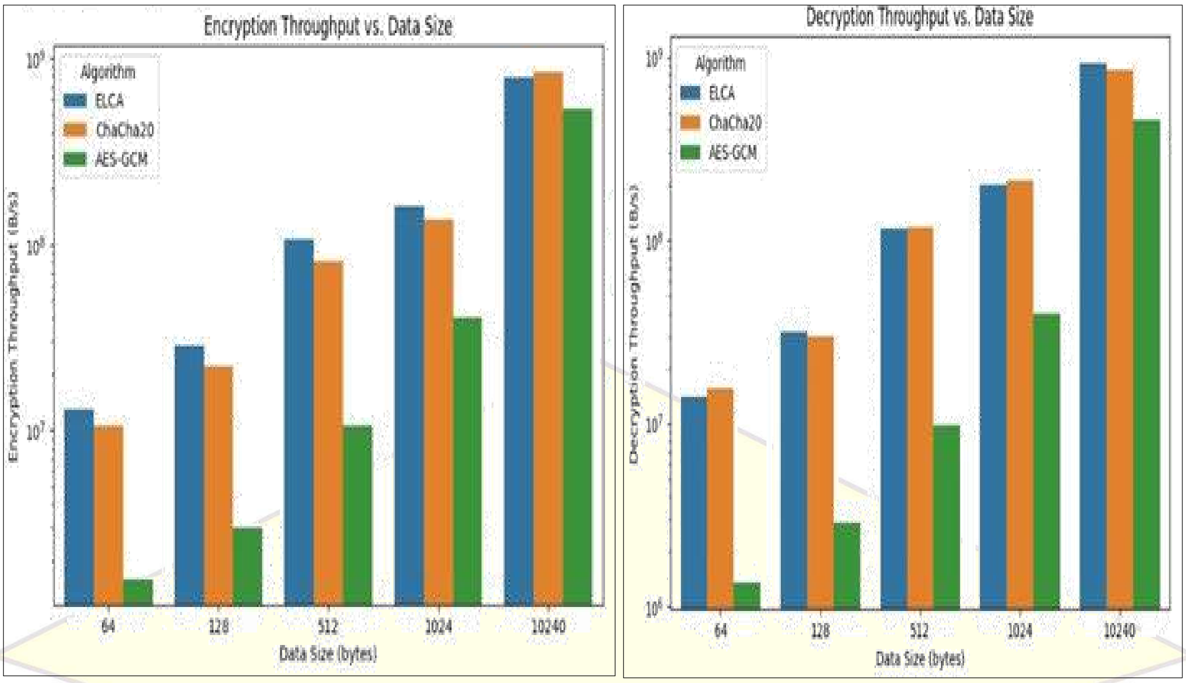

**Fig 6&7 Comparison of encryption/decryption throughput of ELCA, AES-GCM & Chacha20**

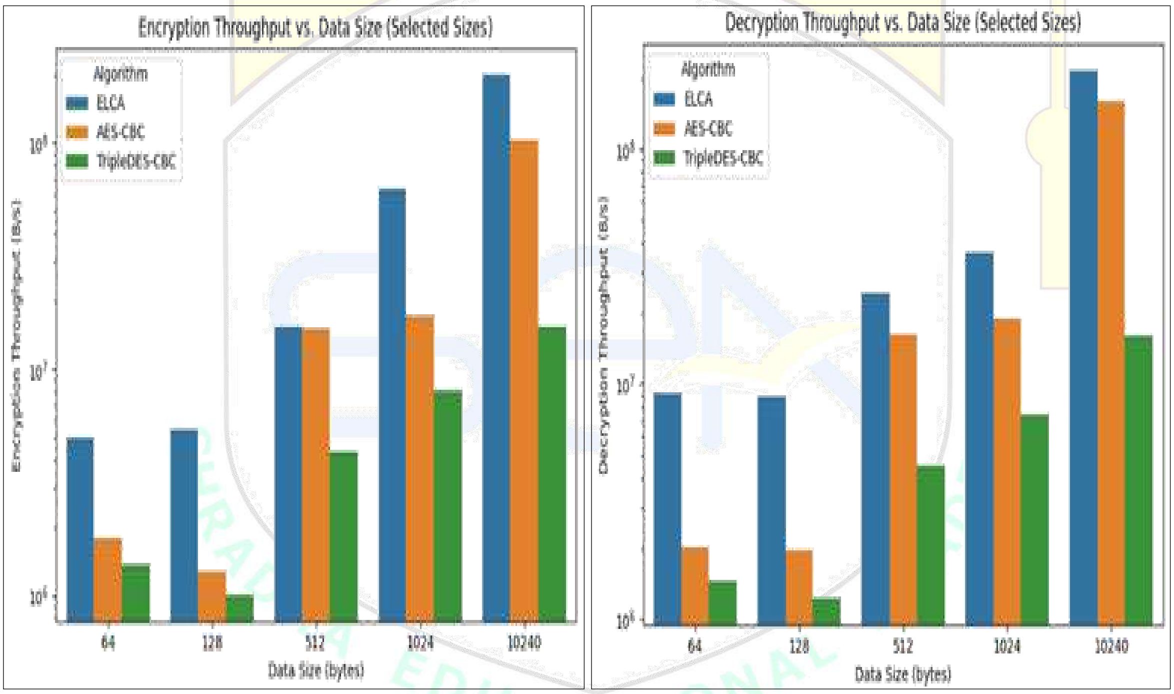

**Fig 8&9 Comparison of encry/decry throughput of ELCA, AES-CBC & TripleDES-CBC**

**6.3 Latency-**Latency refers to the time delay in processing each encryption or decryption operation. Across all datasets, ELCA displayed the lowest encryption and decryption latency, particularly for small and medium-sized data packets. Its minimal time lag indicates efficient algorithmic design. Comparisons with AES-GCM, ChaCha20, and AES-CBC showed that ELCA consistently maintained reduced latency across varying input sizes, ensuring faster response times even under continuous data transmission conditions. The line plots from experimental analysis showed that ELCA's latency increased very gradually with data size, demonstrating stable and predictable performance suitable for real-time IoT systems.

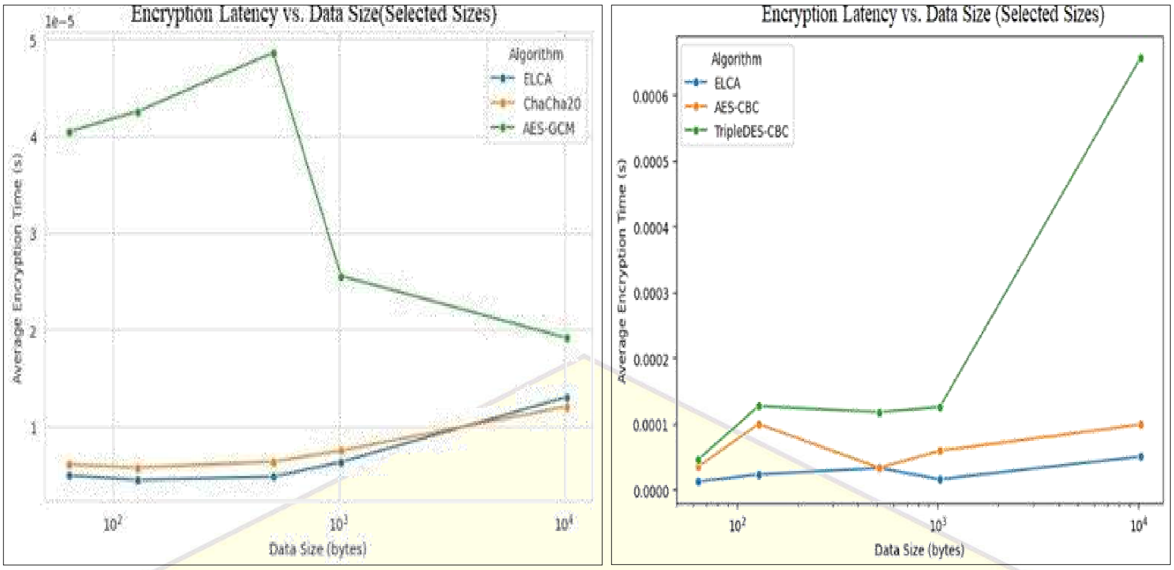

**Fig 10&11 Comparison of encryption latency of ELCA with other standard algorithms**

6.4 Power Consumption

Power consumption was estimated using the measure_power_consumption() function, which analyzed the elapsed time for repeated encryption operations as an indirect measure of energy usage. Results indicate that ELCA consumed the least power among all tested algorithms**,** completing 1000 encryption iterations in just 0.003487 seconds, compared to 0.039553 seconds for Triple DES and higher times for AES variants.

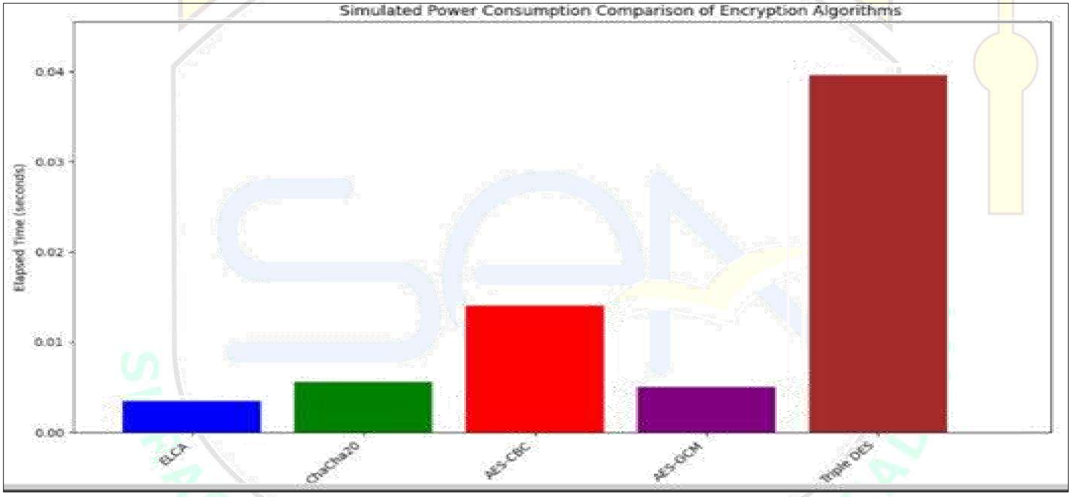

**Fig 12 Power Consumption comparison of ELCA with other standard algorithms**

Since shorter execution durations correlate with lower energy use, ELCA proves to be the most power-efficient algorithm**,** ideal for battery-operated IoT devices where energy conservation is vital. Here also ELCA showed a distinct advantage due to its optimized hybrid design.

The comprehensive analysis confirms that the proposed ELCA algorithm outperforms many standard lightweight cryptographic techniques across nearly all metrics. ELCA demonstrated shortest execution time, highest throughput, minimal latency and lowest power consumption. These results collectively establish ELCA as an ideal choice for lightweight, real-time cryptography in IoT-based healthcare monitoring systems. Furthermore, ELCA's hybrid design provides scalability and adaptability for future hardware acceleration.

**7. Conclusion and Future Scope:-**

The IoT supported systems especially in the healthcare sector face security and performance issues. The designed framework aims at striving a perfect balance between the two. The reduced key size and computational simplicity of ARX design structure is particularly suitable for IoT nodes with limited hardware resources. The integrated authentication and lightweight encryption mechanisms safeguard against common attacks, including data interception, replay, and modification. The model thus provides a secure foundation for IoT-based healthcare systems, enabling continuous and trustworthy monitoring of diabetic patients while preserving data privacy and system reliability. The study thereby contributes to the broader objective of creating secure, scalable, and high-performance frameworks for IoT-based smart healthcare systems, ensuring both patient safety and system reliability. Inculcating genetic algorithm logic with encryption schemes can prove to be a promising approach for future secure healthcare and smart device networks. Prospective research work in this area will aid in bridging the gap between strong encryption and practical performance in real-world IoT environments. The aim should not only be to ensure confidentiality and authenticity but also optimized computational efficiency.

**REFERENCES:**

1. Liu, X., & Baiocchi, O. (2016). A comparison of the definitions for smart sensors, smart objects and Things in IoT. IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 1-4.
2. Yu, T., & Wang, X. (2022). Real-time data analytics in Internet of Things systems. In Handbook of real-time computing (pp. 541-568). Singapore: Springer Nature Singapore.
3. Yang, Y., Wu, L., Yin, G., Li, L., & Zhao, H. (2017). A survey on security and privacy issues in Internet-of-Things. IEEE Internet of things Journal, 4(5), 1250-1258.
4. Alluhaidan, A. S. D., & Prabu, P. (2023). End-to-end encryption in resource-constrained IoT device. IEEE access, 11, 70040-70051.
5. Najm, Z., Jap, D., Jungk, B., Picek, S., & Bhasin, S. (2018, October). On comparing side-channel properties of AES and ChaCha20 on microcontrollers. In 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) (pp. 552-555). IEEE.
6. Elminaam, D. S. A., Kader, H. M. A., & Hadhoud, M. M. (2008). Performance evaluation of symmetric encryption algorithms. *IJCSNS International Journal of Computer Science and Network Security*, *8*(12), 280-286.
7. Haque, M. E., Zobaed, S. M., Islam, M. U., & Areef, F. M. (2018, December). Performance analysis of cryptographic algorithms for selecting better utilization on resource constraint devices. In *2018 21st International Conference of Computer and Information Technology (ICCIT)* (pp. 1-6). IEEE.
8. Valsalan, P., Baomar, T. A. B., & Baabood, A. H. O. (2020). IoT based health monitoring system. Journal of critical reviews, 7(4), 739-743.
9. Mohammed, B. G., & Hasan, D. S. (2023). Smart Healthcare Monitoring System Using IoT. Int. J. Interact. Mob. Technol., 17(1), 141-152.
10. Jadhav, S. P. (2019). Towards light weight cryptography schemes for resource constraint devices in IoT. *Journal of Mobile Multimedia*, 91-110.
11. Al-Husainy, M. A. F., Al-Shargabi, B., & Aljawarneh, S. (2021). Lightweight cryptography system for IoT devices using DNA. *Computers and Electrical Engineering*, *95*, 107418.
12. Mousavi, S. K., Ghaffari, A., Besharat, S., & Afshari, H. (2021). Security of internet of things based on cryptographic algorithms: a survey. Wireless Networks, 27(2), 1515-1555.