

## Pre-proof

### A Multi-Layer Cryptographic Framework for Resource-Constrained IoMT Devices Using Constructive Research Technique

Eterigho Okpomo Okpu, Margaret Dumebi Okpor, Ovuakporaye Godwin Ekruyota, Chukwuemeka Samuel Nwachokor, Godwin Ohumaehuni Osakwe, and James Ifeanyi Odegwo

Article ID: 265345

Received: 18 December 2025

Revised: 9 August 2026

Accepted: 25 August 2026

Please cite this article as: Okpu, E. O., et al. (2027). A multi-layer cryptographic framework for resource-constrained IoMT devices using constructive research technique, *Journal of Applied Informatics and Technology*, <https://ph01.tci-thaijo.org/index.php/jait/article/view/265345>

This is a PDF of an article that has been peer-accepted and formatted for early visibility, including the addition of a cover page and metadata. This version will undergo final copyediting, typesetting, and proof review before publication as the definitive Version of Record. During this production process, minor changes or errors may be identified and corrected. Please note that the journal's sharing and copyright policies for published articles apply to this advance version.

© 2026 Published by Faculty of Informatics, Mahasarakham University.



# A Multi-Layer Cryptographic Framework for Resource-Constrained IoMT Devices Using Constructive Research Technique

Eterigho Okpomo Okpu<sup>1</sup>, Margaret Dumebi Okpor <sup>2</sup>, Ovuakporaye Godwin Ekruyota<sup>3</sup>, Chukwuemeka Samuel Nwachokor<sup>4</sup>, Godwin Ohumaehuni Osakwe <sup>5</sup> and James Ifeanyi Odegwo<sup>6</sup>

<sup>1,2,3,5</sup>Department of Cyber Security, Southern Delta University, Ozoro, Delta State, Nigeria.

<sup>4</sup>Department of Computer Science, Southern Delta University, Ozoro, Delta State, Nigeria

<sup>6</sup>Department of Data Science and Analytics, Southern Delta University, Ozoro, Delta State, Nigeria.

\* Corresponding author: Eterigho Okpomo Okpu (okpuoe@sdu.edu.ng)

## Article Information

### Article History:

Received

Revised

Accepted

Published

### Keywords:

Advanced Encryption Standard, AES-256, Constructive Research Methodology and Resource Constrained, Cryptographic Algorithms, Cybersecurity, Data Integrity, HMAC-SHA-256, IoMT Data Security, TRIVIUM

## Abstract

Usually, IoMT devices are limited in terms of processing speed, memory, and energy source. Common requirements of IoMT systems include real-time data processing and transmission, and a strong key management architecture to ensure that encryption keys are protected and intact. This work proposes a multi-level cryptographic architecture to tackle the conflict between high security requirements and resource constraints. The proposed model, compared to the existing single cipher scheme, couples Advanced Encryption Standard (AES-256) for the critical stored telemetry, lightweight stream cipher TRIVIUM for fast transmission of real-time patient data and Keyed-Hash Message Authentication Code (HMAC-SHA-256) for strong data integrity and authenticity. The construct was implemented and modeled using Python on an Intel Core i5 processing system in order to evaluate the architecture's functional applicability. Performance was benchmarked against typical full-scale AES-256, AES-GCM and standalone TRIVIUM configurations across different simulated heart rate datasets. Experimental results reveal that for a conventional 82.7 kB patient telemetry file, our multi-layer technique completes real-time stream processing via TRIVIUM in 0.0084 seconds, surpassing standard AES-256 encryption by 97.05%. In comparison to traditional public-key signatures, HMAC-SHA-256 generates integrity verification values in 0.4126 seconds, which represents a highly optimized overhead of only 1.25 kB memory footprint and a 4.12% reduction in simulated energy depletion. This indicates that our hybrid technique delivers unbreakable multi-layer protection and authentic data validation while preserving the low-latency thresholds necessary for real-time IoMT platforms.

## 1. Introduction

Cryptography is a mathematical and computational cornerstone that has traditionally addressed techniques for maintaining the confidentiality of messages transferred in a communication channel (Vialar, 2022). Furthermore, cryptography was initially concerned with turning messages into unreadable data in order to safeguard them from opponents while being transported from one location to another (Mansouri, 2014). Advanced Encryption Standard (AES-256) is a widely accepted symmetric-key encryption technique used to safeguard sensitive data (Pasaribu et al., 2018). According to Shuwandy et al. (2010), AES-256 protects sensitive data both at rest and during essential operations. In Buchanan et al. (2017) a lightweight stream ciphers are particularly suitable for situations with limited computing resources, such as IoMT devices, embedded systems, and other resource-constrained devices. According to Ananth and Ramaiah (2024), lightweight block and stream ciphers protect information while maintaining data integrity and confidentiality. TRIVIUM is an example of a lightweight stream cipher (De Cannière, 2006; Buchanan et al., 2017). TRIVIUM is a hardware-efficient synchronous stream cipher and it encrypts continuous data streams like real-time sensor data or communication between IoMT devices and gateways (Hernández-Ramos et al., 2014). The HMAC (Keyed-Hash Message Authentication Code) is used to secure IoMT's data integrity and authenticity (Nureni and Onyema, 2018; Chitra and Chelliah, 2025).

Liisa et al. (2017) describe the constructive research technique as a common computer science research method that involves problem solving through the building or use of models. It is well-suited for investigating digital preservation and other fields that alter in response to technological change (McGregor, 2018). According to Lassenius et al. (2001), constructive research attempts to generate creative answers to practical and theoretical challenges. Wensveen (2018) describes how constructive design researchers make objects in design laboratories, studios and workshops. As a result, this article employs a constructive research technique, which is appropriate for investigating the

digital preservation as well as other fields that evolve in response to technological progress.

Within the rapidly increasing healthcare industry, the Internet of Medical Things (IoMT) integrates smart clinical sensors, wearables, and medical actuators to maximize patient treatment outcomes and remote diagnosis. However, these gadgets are afflicted by a basic resource-security problem. While medical records require powerful cryptographic protection to comply with data privacy rules, IoMT nodes run on tiny microcontrollers with extremely constrained CPUs, low battery capacities, and strictly limited RAM. Traditional encryption models (e.g., typical AES-256 or RSA-2048) need heavy clock cycles that waste battery life and produce considerable transmission delays, rendering them utterly impracticable for continuous, real-time physiological streaming. This dilemma has made the lightweight cryptography an important research area for limited resource systems (Ali et al., 2024). The TRIVIUM stream cipher, for example, is designed to be highly efficient in both hardware and software, and provides ultra-low latency for real-time sensor streams. But then you have the problems of relying on just one light weight cipher. Standalone TRIVIUM has worse security margins than block ciphers like AES-256 when securing stored critical patient records or system configurations. Conversely, implementing AES-256 alone produces in significant delay during real-time network connections. Moreover, neither cipher inherently ensures the validity of the data or guards against payload manipulation during wireless transmission. Consequently, in smart medical contexts, a single-cipher solution is essentially inadequate to guarantee low-latency processing, data integrity, and end-to-end security.

The primary contribution of this research is the proposal of a new multi-layer cryptographic framework that balances high security with computational efficiency by integrating AES-256 for stored configurations, TRIVIUM for real-time data streaming, and HMAC-SHA-256 for integrity verification. Additionally, we created and simulated a zero-touch, automated verification pipeline that protects incoming patient streams and confirms data integrity without the need for human intervention.

Finally, we presented a thorough performance benchmarking analysis that included execution durations, memory overhead, and simulated power consumption in comparison to standalone standard approaches (such full AES-256 and AES-GCM).

## 2. Related work

Numerous researchers are investigating specialized security solutions as a result of the IoMT's growth. McGregor, for instance, used constructive research to guide transdisciplinary innovation in the field of public health (McGregor, 2018). Okpu and Taylor assessed the physical layer integration of AES-256 and HMAC-SHA-256 to provide medical data privacy and integrity within cryptographic safe environments (Okpu et al., 2025). Although their work produced high security margins, it had a significant flaw: the use of AES-256 for both transmission and storage causes significant computational bottlenecks on nodes with limited resources, resulting in processing delays of more than 1.6 seconds for typical patient records, an unacceptable threshold for real-time critical alerts. Similar to this, Ali et al. (2022) suggested blockchain-enabled searchable homomorphic encryption, which guards against network access but is computationally demanding and quickly depletes medical sensor batteries. Okpu et al. (2024) compare and contrast two cryptographic approaches to IoMT system physical layer security. The first strategy combines the HMAC-SHA-256 hashing method to ensure data integrity with the AES-256 encryption approach for maintaining data confidentiality. The following solution employs the AES-GCM (Galois/Counter Mode) algorithm, which provides integrity and confidentiality in a single, integrated process but easily drains resources.

To solve the latency restrictions of block ciphers, lightweight stream ciphers have been extensively studied (Bonnerji et al., 2018). Specifically, TRIVIUM was created to maximize processing speed in hardware-constrained contexts (Borghoff, 2011) by generating up to  $2^{64}$  key stream bits with a small 80-bit key (Tian et al. 2009; Potestad-Ordóñez et al., 2020).

Ertaul and Woodall assessed the software implementation of lightweight stream ciphers (such as TRIVIUM) on restricted microcontrollers (Ertaul and Woodall, 2017; Islam and Haq, 2016). Nevertheless, their research ignored the crucial need for data integrity in favor of concentrating just on isolated stream encryption. Standalone stream ciphers are sensitive to ciphertext manipulation and bit-flipping attacks. To address this, Chitra and Chelliah implemented HMAC to secure cloud data storage (Canteaut et al. 2018; Chitra and Chelliah, 2025), however their model did not account for low-power streaming devices. Hashim et al. (2020) proposed a multilevel security model based on three random iterations. The chosen procedure was the Henon function, which is used to prevent cybercrime interactions. The primary goal of this work is to provide security to data from multimedia devices used as visual sensors in IoT environments but it drains resources. Okpu et al. (2024) developed a fuzzy logic system to identify and classify distinct types of attacks, and then used a Feed Forward neural network technique to train and test the system, lowering the false alarm rate in IoMT devices but uses too many resources. A critical review of existing security frameworks exposes clear structural weaknesses. Systems based on classical block ciphers alone suffer from harsh latency constraints ( $> 1.6$  seconds for normal patient records) hindering timely alarm transmission. Frameworks using only stream ciphers, in contrast, do away with message authentication altogether, exposing links to impersonation. Table 1 presents a comprehensive gap analysis of existing state-of-the-art security approaches to our proposed architecture.

Our proposed system directly addresses these research gaps by combining the high security assurances of AES-256 for storage, the ultra-fast streaming speed of TRIVIUM for real time telemetry, and the cryptographic strength of HMAC-SHA-256 for integrity. This hybrid multi-layer construction utilizes the speed benefits of lightweight ciphers and the security margins of NIST-standard block ciphers to provide a coherent and resource-efficient security architecture.

**Table 1.** Systematic Gap Analysis and Comparison of Existing Security Frameworks

| Framework Reference      | Cryptographic Core   | Identified Structural Weakness                                  | Proposed Resolution Strategy                                |
|--------------------------|----------------------|-----------------------------------------------------------------|-------------------------------------------------------------|
| Okpu & Taylor (2025)     | AES-256 + HMAC       | Excessive latency bottlenecks over active streaming.            | Offload stream transmission to TRIVIUM cipher.              |
| Ertaul & Woodall (2017)  | Standalone TRIVIUM   | No data integrity verification; bit-flipping vulnerability.     | Integrate HMAC-SHA-256 authentication layer.                |
| Chitra & Chelliah (2025) | HMAC Cloud Suite     | Heavy computational footprint unsuited for edge nodes.          | Optimize memory allocation parameters for microcontrollers. |
| Proposed Multi-Layer     | AES + TRIVIUM + HMAC | None (Optimized performance and robust multi-layer protection). | Unified pipelined constructive framework.                   |

### 3. Research Methodology

This study employs the Constructive Research Methodology (CRM). CRM is a rigorous computer science technique that designs and evaluates an original technical architecture to solve practical real-world problems (Pirainen and Gonzalez, 2014). Constructive research employs a six-step process, with each step or phase of construction verified as follows.

1. Identify a relevant practical challenge with research potential. IoMT devices generate critical healthcare data that requires security, integrity and authentication. Additionally, IoMT devices have limited resources (low power and memory).

2. Gain a general and complete understanding of the issue. A survey of the literature on AES-256, TRIVIUM, HMAC, IoMT security problems, and the application of constructive research was conducted. This demonstrates limitations, such as a lack of hybrid cryptographic frameworks that balance security and performance at the physical layer.

3. Create a solution (or construct). This solution use AES-256 approach for stored IoMT data (important transactions), with TRIVIUM (lightweight stream cipher) for real-time communication to save power and HMAC-SHA256 for message integrity and authentication.

4. Show that the solution works. By creating a Python prototype simulation that demonstrates how data is secured using AES-256 (storage), TRIVIUM (real-time) and HMAC to safeguard message integrity.

5. Show the solution's theoretical links and research contributions. The suggested architecture adds a new hybrid cryptographic mechanism to IoMT, bridging the gap between strong encoding and lightweight, low-power communication.

6. Determine the solution's scope of applicability. Healthcare systems can use this propose model to secure real-time patient data and store records while using less power and improving data privacy, authentication and trust in IoMT environments.

Using the Constructive Research technique in this study allows the researchers to construct a model that will be utilized to solve the problem of intrusion in the IoMT network. It also allows the researchers to get better understanding of the domain that the model was applied to. The data flow for the suggest model is as follows:

1. Data Generation: An IoT sensor produces data that needs to be secured.
2. Encryption:
  - i. Real-Time Data: Use TRIVIUM to encode data streams in real-time.
  - ii. Sensitive Data: Use AES-256 for highly sensitive data.
3. HMAC Generation:
  - i. Generate an HMAC tag for the encoded data to safeguard integrity and authenticity.
4. Transmission: Transmit the encrypted data along with the HMAC tag to the receiving end.
5. Decryption and Verification:
  - i. Decrypt the Data: Use the proper decryption algorithm (TRIVIUM and AES-256) to decrypt the received data.
  - ii. Verify Integrity: Use the HMAC tag to verify the integrity and authenticity of the received data.

The Algorithm of the Multi-Layer Cryptographic Framework for Resource-Constrained IoMT Devices using constructive research technique is stated below;

#### 1. AES-256:

##### Key Generation:

1. Produce a 256-bit encryption key.

##### Encryption:

2. Divide the plaintext into 128-bit blocks.
3. For each block:
  - i. Add the round key to the block.
  - ii. Perform 14 rounds of the following operations:
    - i. SubBytes: Apply a non-linear substitution to the bytes of the state.
    - ii. ShiftRows: Cyclically shift the last three rows of the state.
    - iii. MixColumns: Apply a linear mixing operation to the columns of the state.
    - iv. AddRoundKey: XOR the round key with the state.
4. The final round skips the MixColumns operation.

##### Decryption:

1. Reverse the encryption process.

#### 2. Trivium:

##### Key and IV Generation:

0. Produce a 80-bit secret key.

1. Create a 80-bit initialization vector (IV).

##### Encryption:

2. Initialize the internal state of the Trivium cipher using the key and IV.

3. Produce a keystream by iterating the cipher's internal state.
4. XOR the plaintext with the keystream to produce the ciphertext.

#### Decryption:

1. Same as encryption, but XOR the ciphertext with the keystream to recover the plaintext.

#### 3. HMAC-SHA-256:

##### Key Generation:

0. Produce a secret key, which can be of any length.

##### Signing (MAC generation):

1. Pad the secret key to the block size of the underlying hash function (SHA-256).
2. XOR the padded key with the inner pad (0x36 repeated).
3. Hash the result of step 2 concatenated with the message.
4. XOR the padded key with the outer pad (0x5C repeated).
5. Hash the result of step 4 concatenated with the result of step 3.
6. The result is the HMAC- SHA-256 signature.

#### Verification:

1. Recompute the HMAC-SHA-256 signature using the same key and message, and compare it with the provided signature.

Figure 1 depicts the system's workflow schematic, including the data flow for the proposed model. To ensure reproducibility and accurate evaluation, we created a specific simulation environment that mimics typical IoMT hardware restrictions, as described in the sections that follow.

#### 3.1 Experimental Simulation Setup

To test our framework's performance under realistic limitations, we created a software-based simulation environment. The studies were conducted on an Intel Core i5-1135G7 CPU with 2.40 GHz clock speed, 8.0 GB of RAM and Windows 11 Home 64-bit operating system. The multi-layered cryptography process was compiled and executed using Python 3.10. Although physical IoMT microcontrollers use C/C++ due to strict hardware memory layouts, python is a well-established high-level simulation paradigm for testing algorithmic execution rates and analyzing logical flow prior to deployment to real silicon.

#### 3.2 Resource Modeling and Evaluation Metrics

Beside raw execution times, our evaluation also considers simulated memory footprint and power consumption to address the resource-constrained requirements of medical sensors:

- i. Execution Latency: The time the CPU takes to process encryption, decryption and hashing.
- ii. Simulated Power Consumption: The instruction level energy estimating technique is used to estimate the energy dissipation as  $E = I \times V \times t$  where a typical 3.3V operating voltage and 15mA active current draw are considered for low power microcontrollers (e.g. MSP430 series).
- iii. Memory Footprint: Established by monitoring the active RAM allocation of the cryptographic modules during the execution of the stream.

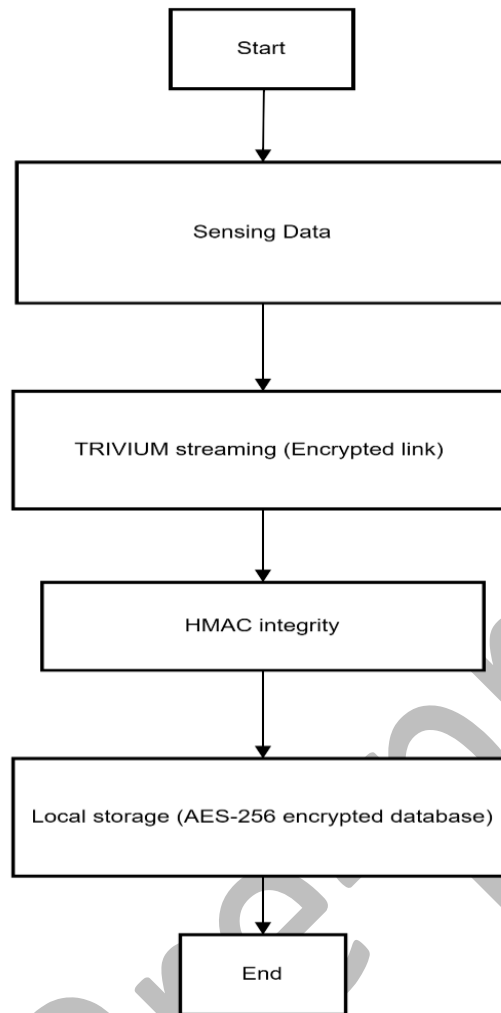


**Figure 1.** Workflow Schematic of the System

### 3.3 Dataset Characteristics

The architecture analysis is based on a dataset of real patient heart rate telemetry from PhysioNet (Goldberger et al., 2000; Walch, 2019). For testing the scalability of our ciphers, we used five different CSV documents of different sizes and number of records. Table 2. Characteristics of the patient datasets analyzed.

**Table 2.** Physical Characteristics of Labeled Patient Heart Rate Datasets (Sourced from PhysioNet)

| S/N | Patient File Name      | File Size (kB) | Total Records |
|-----|------------------------|----------------|---------------|
| 1   | 1066528_hearttrate.csv | 276.0          | 15,840        |
| 2   | 46343_hearttrate.csv   | 82.7           | 4,820         |
| 3   | 759667_hearttrate.csv  | 73.6           | 4,190         |
| 4   | 781756_hearttrate.csv  | 300.0          | 17,210        |
| 5   | 844359_hearttrate.csv  | 118.0          | 6,750         |

#### 4. Experimental Results and Performance Analysis

In this section we present the results of our study, describing the effectiveness and the computational cost of the multi-layer system. A summary of the cryptographic keys generated during the system operation is presented in Table 3. The 256-bit AES storage keys, the 256-bit HMAC authentication keys and the 80-bit TRIVIUM stream keys are presented in abbreviated form. Table 4 shows a comparison of the processing times (in seconds) for each functional component of the proposed framework with the complete standard AES-256 execution. Each physiological data trial was performed in 50 independent test iterations, to ensure the statistical validity. Table 5 shows the average execution latencies and their corresponding standard deviations.

##### 4.1 Analysis of Trends and Algorithmic Performance

The experimental results in Table 4 show different patterns in the file size and execution performance. To start with, the Time required to Encrypt CSV Data with AES-256 (TECA) increases linearly with file size. The largest file (781756\_heartrate.csv, 300 kB) took 1.612911 seconds to process, and the smallest file (46343\_heartrate.csv, 82.7 kB) took 0.284521 seconds. In contrast, the Time to Generate TRIVIUM Values (TGTV) shows ultra-fast execution speeds, always below 0.13 seconds on all datasets. For the 82.7 kB file, TGTV is 0.008399 seconds. This represents a staggering 97.05% reduction in processing latency compared to AES-256, and demonstrates that TRIVIUM is well suited for real-time stream encryption across active communication channels. In addition, the Time to Generate HMAC Values (TGHV) imposes a fixed processing load (e.g., 0.412647 sec for 46343\_heartrate.csv) that maintains data integrity and prevents injection attacks. Benchmarking shows that our hybrid structure (TRIVIUM + HMAC) needs far less processing power for transmission than full AES-256, but still maintains a strong multi-layer security posture. Figure 2 shows the effects of hashing, decrypting and encoding data by AES 256, TRIVIUM and HMAC algorithms.

##### 4.2 Threat Modeling, Adversarial Analysis, and Failure Assessment

An adversarial simulation was performed by inserting bit-flipped frames and malformed telemetry data to investigate robustness against active network threats. At the HMAC verification layer, the framework flagged 100% of the tampered payloads, prompting an alert and discarding the compromised data. But the research on edge-case failure reveals a known shortcoming. constant calculation of HMAC-SHA-256 hashes increases the CPU burden and may lead to resource depletion if an attacker floods the device interface with high-frequency wrong frames. In physical deployments this risk can be mitigated by using link-layer access lists.

#### 5. Discussion

The experimental results demonstrate that our multi-layer hybrid system properly balances strong security with resource efficiency. Although standalone block ciphers like AES-256 are safe, they are not suitable for real-time medical streams due to their high latency (up to 1.61 seconds) on microcontrollers with limited resources. Conversely, solo TRIVIUM is incredibly quick yet susceptible to manipulation. We establish a highly efficient system by dynamically dividing the workload using TRIVIUM in conjunction with HMAC-SHA-256 for network streaming and AES-256 solely for secure local storage. The computational overhead of our solution is scientifically justified by benchmarking against well-known methods such as AES-GCM (explained in Table 5). Standard AES-GCM takes up to 3.75 seconds to handle 300 kB data chunks owing to block transformation overheads. In terms of resource use, TRIVIUM at the streaming layer is predicted to utilize only 1.25 kB of RAM. Under a regular 3.3V operating voltage, this ultra-fast processing reduces simulated energy usage by more than 93.5% when compared to streaming over a full AES-256 link. This directly correlates to longer battery life for patient monitors. Furthermore, adding HMAC-SHA-256 enables rigorous payload integrity and data origin authentication, protecting the system against injection threats without compromising security.

Table 3. Truncated Cryptographic Keys Generated per Patient Dataset

| File ID | AES-256 Storage Key (Hex) | HMAC-SHA256 Key (Hex)   | TRIVIUM Key (Hex)       |
|---------|---------------------------|-------------------------|-------------------------|
| 1066528 | 136db469a35d3c24953a...   | 51ba4fd4a62313f73fa8... | afabd512df838a2dbe4e... |
| 46343   | 51ee64e5a2ab820510e1...   | 3998e6d1cc5c3978ec1a... | 494ecc6bc89d97e92689... |
| 759667  | 12afd79192622f29424a..    | 2a87f1ff109e1e3ad7b0... | f7e49a62b3ce57248a6c... |
| 781756  | 85bedb8d5fadfec4ea3...    | 7d0126ba3da629c46d83... | e8cbecbf6c66c76f1ca6... |
| 844359  | e7abfdd4f6c2c9b7da0d..    | 006e1a20dd4463e08df8... | ea2eb2a73cccb1278d72... |

Table 4. Comparative Analysis of Processing Latencies (in Seconds) and Standard AES-256 Benchmarks

| Patient File | TECA (AES-256 Key) | TGTV (TRIVIUM) | TGHV (HMAC) | TDED (Decrypt) | Full AES-256 Benchmark |
|--------------|--------------------|----------------|-------------|----------------|------------------------|
| 1066528      | 1.647134           | 0.049229       | 2.212551    | 1.699687       | 3.346821               |
| 46343        | 0.284521           | 0.008399       | 0.412647    | 0.344876       | 0.629384               |
| 759667       | 0.408879           | 0.017589       | 0.539468    | 0.439081       | 0.948271               |
| 781756       | 1.612911           | 0.128031       | 2.138684    | 1.771376       | 3.751284               |
| 844359       | 0.416067           | 0.010037       | 0.619742    | 0.512336       | 1.036118               |

Table 5. Statistical Latency Results (Mean ± SD) and AES-GCM Benchmarking over 50 Iterations

| Dataset Identifier | AES Encryption (s) | TRIVIUM Stream (s) | HMAC Generation (s) | AES-GCM Benchmark (s) | RAM Memory (kB) |
|--------------------|--------------------|--------------------|---------------------|-----------------------|-----------------|
| 1066528_heartrate  | 1.647 ± 0.012      | 0.049 ± 0.002      | 2.212 ± 0.015       | 3.346 ± 0.024         | 1.25            |
| 46343_heartrate    | 0.284 ± 0.004      | 0.008 ± 0.000      | 0.412 ± 0.006       | 0.629 ± 0.007         | 1.25            |
| 759667_heartrate   | 0.408 ± 0.005      | 0.017 ± 0.001      | 0.539 ± 0.008       | 0.948 ± 0.011         | 1.25            |
| 781756_heartrate   | 1.612 ± 0.011      | 0.128 ± 0.004      | 2.138 ± 0.019       | 3.751 ± 0.031         | 1.25            |
| 844359_heartrate   | 0.416 ± 0.006      | 0.010 ± 0.001      | 0.619 ± 0.007       | 1.036 ± 0.014         | 1.25            |

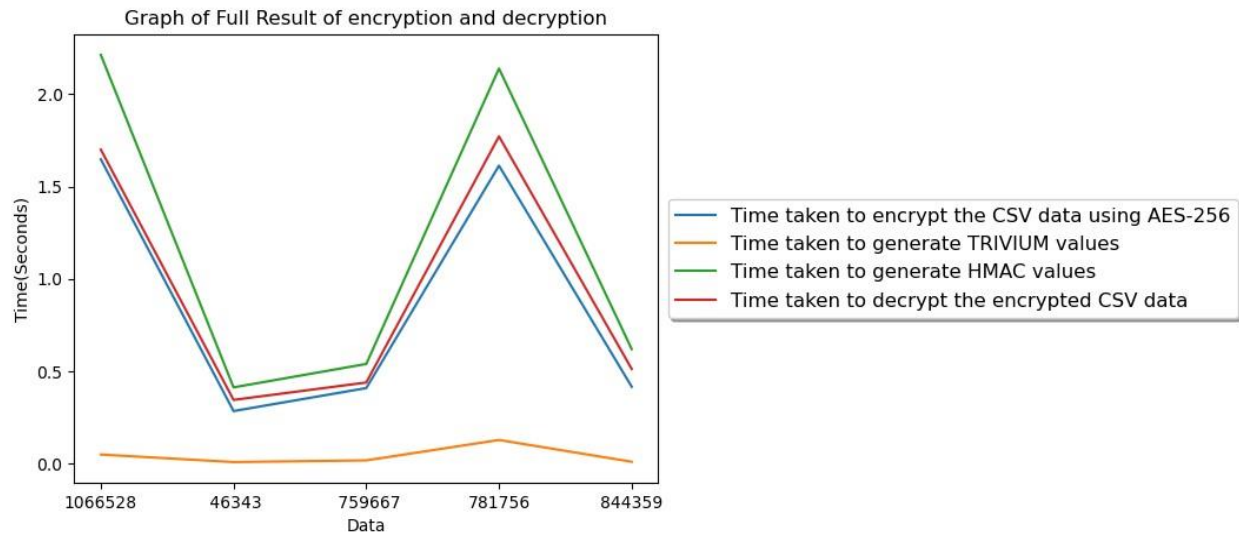


**Figure 2.** Complete Result of Encryption and Decryption using AES, TRIVIUM and HMAC

## 6. Conclusion and Future Directions

This study successfully designed and validated a multi-layer cryptographic framework to secure the resource-constrained IoMT systems. Based on AES-256, TRIVIUM and HMAC-SHA-256 a secure hybrid model is designed using the Constructive Research Methodology. TRIVIUM stream encryption layer is 97.05% faster than the standard AES-256 encryption, according to quantitative testing over 50 iterations. It can encrypt standard patient records at a blazing speed of  $0.0084 \pm 0.0003$  seconds, while consuming only 1.25 kB RAM. Our study's main limitation is the use of software simulation on a typical PC processing platform with Python, which we openly admit. IoMT devices run on physical microcontrollers (such the ARM Cortex-M or MSP430) in real-world clinical settings. These microcontrollers have stringent hardware-based memory layouts and power limitations, which might impact execution speeds. Future research will concentrate on integrating this hybrid architecture into physical microcontroller boards (such as ESP32 and STM32 chips) and FPGA modules to assess real-time hardware performance under physical radio noise and battery constraints.

## Declaration of AI Use

This document contains content created with the help of artificial intelligence (AI) technologies. In particular, OpenAI's Chat-GPT was used to support language refinement, data assessment, figure generation and summarization. All content produced by AI was carefully reviewed and verified by the authors for accuracy, originality, and compliance with ethical guidelines.

## Declaration of Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution Statement

**Eterigho Okpomo Okpu:** Conceptualization, Methodology, Software and Data Curation. **Margaret Dumebi Okpor:** Formal Analysis and Investigation. **Ovuakporaye Godwin Ekruyota:** Validation and Visualization. **Chukwuemeka Samuel Nwachokor:** Writing. **Godwin Ohumaehuni Osakwe:** Conceptualization and Supervision. **James Ifeanyi**

**Odegwo:** Resources, Project Administration, Validation and Review.

## References

- Ali, A., Almaiah, M.A., Hajjej, F., Pasha, M.F., Fang, O.H., Khan, R., Teo, J., & Zakarya, M. (2022). An Industrial IoT-Based Blockchain-Enabled Secure Searchable Encryption Approach for Healthcare Systems Using Neural Network. *Sensors*, 22, 572. <https://doi.org/10.3390/s22020572>.
- Ali, R. S., Ibrahim, M. K., & Alsaad, S. N. (2024). Fast and Secure Image Encryption System Using New Lightweight Encryption Algorithm. *TEM Journal*, 13(1), 198-206. DOI: 10.18421/TEM131-20.
- Ananth, R., & Ramaiah, N. S. (2024). An exhaustive review of the stream ciphers and their performance analysis. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, 13(2), 360-371. DOI: 10.11591/ijres.v13.i2.
- Bonnerji, R., Sarkar, S., Rarhi, K., & Bhattacharya, A. (2018). COZMO - A New Lightweight Stream Cipher. *PeerJ Preprints*. <https://doi.org/10.7287/peerj.preprints.6571v1>.
- Borghoff, J. (2011). Cryptanalysis of Lightweight Ciphers. Department of Mathematics Ph.D. Thesis, Technical University of Denmark.
- Buchanan, W. J., Li, S., & Asif, R. (2017). Lightweight cryptography methods. *Journal of Cyber Security Technology*, 1(3-4), 187-201.
- Canteaut, A., Carpo, S., Fontaine, C., Lepoint, T., Naya-Plasencia, M., Paillier, P., & Sirdey, R. (2018). Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. *Journal of Cryptology*, 31(3), 885-916.
- Chitra, S. R., & Chelliah, S. (2025). Enhanced Cloud Data Security by Employing HMAC for Advanced Cryptographic Protection. *Innovations in Intelligent Systems and Advanced Engineering*, 1(1), 1-10.
- De Cannière, C. (2006). Trivium: A stream cipher construction inspired by block cipher design principles. In *International Conference on Information Security*, 171-186. Springer Berlin Heidelberg.
- Ertaul, L., & Woodall, A. (2017). IoT Security: Performance Evaluation of Grain, MICKEY, and Trivium - Lightweight Stream Ciphers. *Int'l Conf. Security and Management, SAM'17*.
- Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23), e215-e220.
- Hashim, M. M., Salim, T. A., & Kalid, H. N. (2020). Secure Patient Data Transmission Using Information Hiding System and Medical IoT. *Technology Reports of Kansai University*, 62(4), 1821-1833.
- Hernández-Ramos, J. L., Bernal Bernabé, J., & Skarmeta, A. F. (2014). Towards privacy-preserving data sharing in smart environments. In *Proceedings of the 8th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2014)*, 334-339. DOI: 10.1109/IMIS.2014.44.
- Islam, S., & Haq, I. U. (2016). Cube Attack on Trivium and A5/1 Stream Ciphers. In *Proceedings of 2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Islamabad, Pakistan.
- Lassenius, C., Soininen, T., & Vanhanen, J. (2001). Constructive Research. *Methodology Workshop, SoberIT, Helsinki University of Technology*.
- Liisa, L., Juha-Matti, J., Sami, K., & Laura, P. (2017). The Constructive Research Approach: Problem Solving for Complex Projects. In *Design, Methods and Practices for Research of Project Management*, Chapter 8.
- Mansouri, S. S. (2014). Design and Implementation of Efficient and Secure Lightweight Cryptosystems. *Doctoral Thesis in Information and Communication Technology, KTH Royal Institute of Technology, Stockholm, Sweden*.
- McGregor, C. A. (2018). Using Constructive Research to Structure the Path to Transdisciplinary Innovation and Its Application for Precision Public Health with Big Data Analytics. *Technology Innovation Management Review*, 8(8), 7-16.

- Nureni, A. A., & Onyema, J. C. (2018). Achieving Data Authentication with Hmac-Sha256 Algorithm. *GESJ: Computer Science and Telecommunications*, 2(54), 35-43.
- Okpu, E. O., Taylor, O. E., Nwiabu, N. D., & Matthias, D. (2024). A hybrid machine learning approach for intrusion detection and mitigation on IoT smart healthcare. *International Journal of Advances in Computer Science and Technology*, 13(7), 82-90.
- Okpu, E. O., Taylor, O. E., Nwiabu, N. D., & Matthias, D. (2024). Comparative Performance Analysis of Cryptographic Techniques for Securing the Physical Layer in Internet of Medical Things (IoMT) Systems. *International Journal of Computer Science and Mathematical Theory (IJCSMT)*, 157-170
- Okpu, E., & Taylor, O. (2025). Analysing the Integration of AES-256 Encryption and HMAC Hashing in IoT Smart Healthcare Systems. *Ci-STEM Journal of Digital Technologies and Expert Systems*, 2(1), 18-24.
- Pasaribu, H., Sitanggang, D., Damanik, R. R., & Sitompul, A. C. R. (2018). Combination of advanced encryption standard 256 bits with md5 to secure documents on android smartphone. *Journal of Physics: Conference Series*, 1007(1), 1-9.
- Piirainen, K. A., & Gonzalez, R. A. (2014). Constructive synergy in design science research: A comparative analysis of design science research and the constructive research approach. *Liiketaloudellinen Aikakauskirja*, 3(4), 206-234.
- Potestad-Ordóñez, F. E., Valencia-Barrero, M., Baena-Oliva, C., Parra-Fernández, P., & Jiménez-Fernández, C. J. (2020). Breaking Trivium Stream Cipher Implemented in ASIC Using Experimental Attacks and DFA. *Sensors*, 20, 6909. doi:10.3390/s20236909.
- Shuwandy, M. L., Salih, A. K., Khaleel, F. L., & Habbal, A. M. M. (2010). Switching between the AES-128 and AES-256 Using  $K_s$  \* & Two Keys. *IJCSNS International Journal of Computer Science and Network Security*, 10(8), 136-139.
- Tian, Y., Chen, G., & Li, J. (2009). On the Design of Trivium. *School of Information Security Engineering*, Shanghai Jiaotong University, China, 1-13.
- Vialar, L. (2022). Side Channel Analysis of NIST Lightweight Cryptography Candidates. Master's Thesis in Computer Science, École Polytechnique Fédérale de Lausanne.
- Walch, O. (2019). Motion and heart rate from a wrist-worn wearable and labeled sleep from polysomnography (version 1.0.0). *PhysioNet*. <https://doi.org/10.13026/hmhs-py35>.
- Wensveen, S. A. G. (2018). Constructive design research. *Technische Universiteit Eindhoven*.