Journal of Computer Science and Technological Enquiry 2 (02) (2026)

Contents lists available at Noesis Publishing

Journal of Computer Science and Technological Enquiry

journal homepage: **https://journals.noesispub.com/jcste**



Research paper

DMSA-Based Secure IoT Communication System Using ESP32 and DHT11 Sensor

Kumari Puspo Rani[a1*], Md Younus[a], Tayeb Hassan[a], Redwanul Islam Rabbi[a]

[a] *Department of Computer Science and Engineering, Jamalpur Science and Technology University, Bangladesh*

ARTICLE INFO

Keywords:
IoT Security
ESP32
DMSA
Encryption
Secure Communication

ABSTRACT

The rapid expansion of Internet of Things (IoT) systems has increased the need for secure real-time sensor communication. However, many lightweight IoT devices suffer from limited security due to low computational resources. This research proposes a lightweight secure IoT communication framework using ESP32 and DHT11 sensors integrated with a custom Dynamic Matrix Shift Algorithm (DMSA). The system collects real-time temperature and humidity data from the DHT11 sensor through ESP32 and securely transmits the data between sender and receiver applications. The proposed DMSA performs multiple encryption stages including ASCII conversion, binary transformation, matrix generation, row shifting, column swapping, and binary rotation to generate encrypted cipher text. The receiver application applies reverse DMSA operations to recover the original sensor data. Experimental implementation demonstrates that the proposed system successfully secures real-time IoT sensor communication while maintaining low computational complexity suitable for lightweight embedded systems.

1. Introduction

The Internet of Things (IoT) has become an important component of modern communication and cyber-physical systems, enabling interconnected devices to sense, process, and exchange information across applications such as smart homes, healthcare, agriculture, industrial automation, and environmental monitoring. As the number of connected devices continues to increase, the security and privacy of transmitted sensor data have become important research concerns. In particular, IoT devices are often characterized by limited computational capability, memory, energy, and communication resources, which makes the deployment of computationally intensive security mechanisms challenging [1], [2].

Lightweight cryptography has therefore emerged as an important research direction for securing resource-constrained IoT environments. Recent research has investigated lightweight cryptographic approaches designed to provide security while reducing computational and resource requirements. Rana et al. reviewed lightweight cryptographic algorithms for IoT networks and highlighted the importance of developing security mechanisms that consider the resource limitations of IoT devices [1]. Similarly, a comprehensive survey of lightweight block ciphers demonstrated that security, computational performance, and resource consumption must be considered jointly when designing cryptographic solutions for constrained environments [2]. These studies indicate that the suitability of a cryptographic mechanism for an IoT application cannot be determined from security characteristics alone; its practical implementation cost must also be considered.

[1] Corresponding author: Kumari Puspo Rani.
E-mail address: puspod0@gmail.com
https://doi.org/10.65496/jcste.113
Received 06 September 2026; Received in revised form 10 November 2026; Accepted 12 November 2026
Available online: 20 November 2026
3135-0372/© 2026 The Authors. Published by Noesis Publishing. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Several recent studies have experimentally evaluated lightweight cryptographic algorithms on resource-constrained IoT platforms. Radhakrishnan et al. compared AES-128, SPECK, and ASCON using execution time, memory utilization, latency, throughput, and security-related characteristics on constrained IoT boards [3]. Their results demonstrate the importance of benchmarking cryptographic mechanisms using both security and performance metrics. Alcaraz-Velasco et al. proposed GS3, a lightweight method based on data-block generation, shuffling, scrambling, and substitution for constrained IoT devices, and experimentally evaluated its execution and communication overhead [4]. Such studies demonstrate that practical evaluation on embedded platforms is important when assessing the suitability of lightweight security mechanisms for IoT communication.

Recent literature has also emphasized that the evaluation of lightweight cryptographic mechanisms should consider multiple security and performance characteristics rather than relying only on successful encryption and decryption. Ansari and Ali conducted a systematic review of 77 peer-reviewed studies published between 2006 and 2025 and identified metrics such as entropy, execution time, energy consumption, throughput, NPCR, and UACI as important measures for evaluating lightweight cryptographic schemes in IoT environments [5]. Their review also highlighted the need for practical validation on real IoT hardware, since a substantial portion of existing research relies on simulation or software-based evaluation. Therefore, an IoT security mechanism should ideally be assessed in terms of computational cost, memory requirements, communication overhead, and relevant statistical security characteristics in addition to functional correctness.

Although established cryptographic algorithms such as the Advanced Encryption Standard (AES) provide strong security, the selection of an encryption mechanism for a constrained IoT application requires consideration of the available computational and memory resources as well as the required security properties [3]. Recent lightweight cryptography research has consequently focused on achieving an appropriate balance between security and resource consumption. In this context, direct experimental comparison with established lightweight cryptographic schemes is important when introducing a new lightweight approach.

Motivated by these considerations, this study investigates a Dynamic Matrix Shift Algorithm (DMSA) for lightweight protection of IoT sensor communication. The proposed framework integrates an ESP32 microcontroller, a DHT11 temperature and humidity sensor, and web-based sender and receiver applications developed using Python Flask. In the implemented prototype, sensor data are transformed through a sequence of matrix-based operations, including ASCII conversion, binary representation, row shifting, column swapping, and binary rotation, before transmission. At the receiver side, the corresponding inverse operations are applied to recover the original sensor data.

The objective of this study is not only to demonstrate the functional encryption and decryption of sensor messages but also to empirically investigate the computational and statistical characteristics of the implemented DMSA approach. The evaluation considers plaintext sensitivity, key sensitivity, Shannon entropy, adjacent-byte correlation, and execution-time performance. Communication characteristics are also considered to examine the overhead introduced during transmission. Where applicable, the performance and characteristics of DMSA are compared with existing lightweight cryptographic approaches to assess its practical suitability and limitations for resource-constrained IoT communication.

The main contributions of this study are summarized as follows:

- A Dynamic Matrix Shift Algorithm (DMSA) is developed as a lightweight matrix-based data transformation approach for protecting IoT sensor data during communication.
- An end-to-end IoT prototype integrating an ESP32 microcontroller, DHT11 temperature and humidity sensor, sender application, receiver application, encryption, transmission, decryption, and data visualization is implemented.
- The implemented DMSA approach is experimentally evaluated using plaintext sensitivity, key sensitivity, Shannon entropy, adjacent-byte correlation, and execution-time measurements.
- Computational and communication characteristics are analyzed to investigate the lightweight properties and practical overhead of the proposed approach.
- The limitations of the current DMSA design are explicitly identified, and its characteristics are discussed in relation to existing lightweight cryptographic approaches.

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 describes the data and methodology, Section 4 presents the results, Section 5 discusses the findings, and Section 6 concludes the paper with directions for future work.

2. Literature Review

Secure communication in Internet of Things (IoT) environments has received considerable research attention because the increasing number of connected devices has created new requirements for protecting sensor data while maintaining low computational and communication overhead. Unlike conventional computing systems, many IoT devices operate with limited processing capability, memory, energy, and storage resources. Consequently, lightweight security mechanisms have been investigated to achieve an appropriate balance between security and resource efficiency [2]. Several lightweight cryptographic algorithms have been developed specifically for constrained environments. Lightweight block ciphers aim to reduce implementation complexity, memory requirements, and computational cost while maintaining appropriate security properties. Zhong and Gu provided a comprehensive survey of lightweight block ciphers for resource-constrained environments and discussed the trade-offs among security, performance, and resource consumption [2]. Their study demonstrates that the design of lightweight cryptographic mechanisms requires simultaneous consideration of security strength and implementation efficiency. Experimental evaluation on resource-constrained hardware has also become an important aspect of recent IoT security research. Radhakrishnan et al. evaluated AES-128, SPECK, and ASCON on resource-constrained IoT platforms using execution time, memory utilization, latency, throughput, and security-related characteristics [3]. Their findings demonstrate that cryptographic algorithms should be assessed using practical performance measurements in addition to their theoretical security properties. Such evaluation provides a more realistic understanding of whether a security mechanism is suitable for embedded IoT applications. Another recent approach is GS3, proposed by Alcaraz-Velasco et al., which was designed for constrained IoT devices using data-block generation, shuffling, scrambling, and substitution operations [4]. The study experimentally investigated the computational and communication characteristics of the proposed method. The approach demonstrates that lightweight data-protection mechanisms can be constructed using relatively simple operations while considering the limited resources of IoT devices. However, the transformation structure of GS3 differs from the matrix-based transformation sequence investigated in the present study. Recent systematic research has further emphasized the importance of multidimensional evaluation of lightweight cryptographic mechanisms. Ansari and Ali reviewed 77 peer-reviewed studies on lightweight cryptographic schemes for IoT security and privacy and identified execution time, energy consumption, throughput, entropy, NPCR, UACI, and other statistical characteristics as important evaluation measures [5]. Their review highlights that successful encryption and decryption alone are insufficient to establish the effectiveness of a lightweight security mechanism. Practical resource consumption and statistical characteristics should also be investigated. In addition to individual cryptographic algorithms, recent research has examined lightweight security from a broader IoT-edge perspective. Gușiță et al. reviewed lightweight cryptography and related security mechanisms for IoT-edge environments and emphasized the need to balance security requirements with the limited computational and communication resources of edge and IoT devices [6]. These studies indicate that modern IoT security research increasingly focuses on practical implementation, resource efficiency, and measurable security characteristics rather than relying solely on conventional encryption demonstrations. Based on the existing literature, established lightweight cryptographic schemes such as SPECK and ASCON are designed as dedicated cryptographic primitives with defined security objectives, while other lightweight approaches employ combinations of data transformation, scrambling, substitution, or permutation operations [3], [4]. The proposed Dynamic Matrix Shift Algorithm (DMSA) follows a different transformation structure based on ASCII conversion, binary representation, matrix construction, row shifting, column swapping, and binary rotation. Therefore, the primary methodological distinction of DMSA is its matrix-oriented transformation sequence integrated directly into an ESP32-based IoT sensor communication framework.

However, the reviewed literature also indicates that a custom transformation sequence should not automatically be considered equivalent to a standardized cryptographic primitive. A new lightweight approach must be evaluated using appropriate security and performance metrics before making strong security claims [2], [3], [5]. In particular, computational overhead, memory consumption, communication overhead, ciphertext characteristics, and statistical properties should be considered when assessing its practical suitability. Therefore, a research gap remains in the practical evaluation of custom matrix-based data transformation approaches within real-time IoT sensor communication systems. While previous studies have investigated standardized lightweight algorithms and other lightweight data-protection techniques, limited work has examined a matrix-oriented transformation approach implemented as an end-to-end ESP32–DHT11 communication framework and evaluated

using both security-related and performance-related measurements. This study addresses this gap by experimentally investigating DMSA using plaintext sensitivity, key sensitivity, Shannon entropy, adjacent-byte correlation, execution time, and communication characteristics. The results are used to identify both the practical advantages and the limitations of the proposed approach.

3. Data and Methodology

The proposed system was designed to establish secure real-time communication between IoT devices using a lightweight custom encryption framework. The system consists of ESP32 microcontrollers, DHT11 sensors, sender applications, receiver applications, and the proposed Dynamic Matrix Shift Algorithm (DMSA).

The overall workflow of the proposed system begins with the collection of real-time temperature and humidity data from the DHT11 sensor through ESP32. The collected sensor data is transmitted through serial communication to the sender application. The sender application applies the DMSA encryption process and securely transmits the encrypted data to the receiver application. The receiver performs reverse DMSA operations to recover the original sensor information, as illustrated in Figure 1.

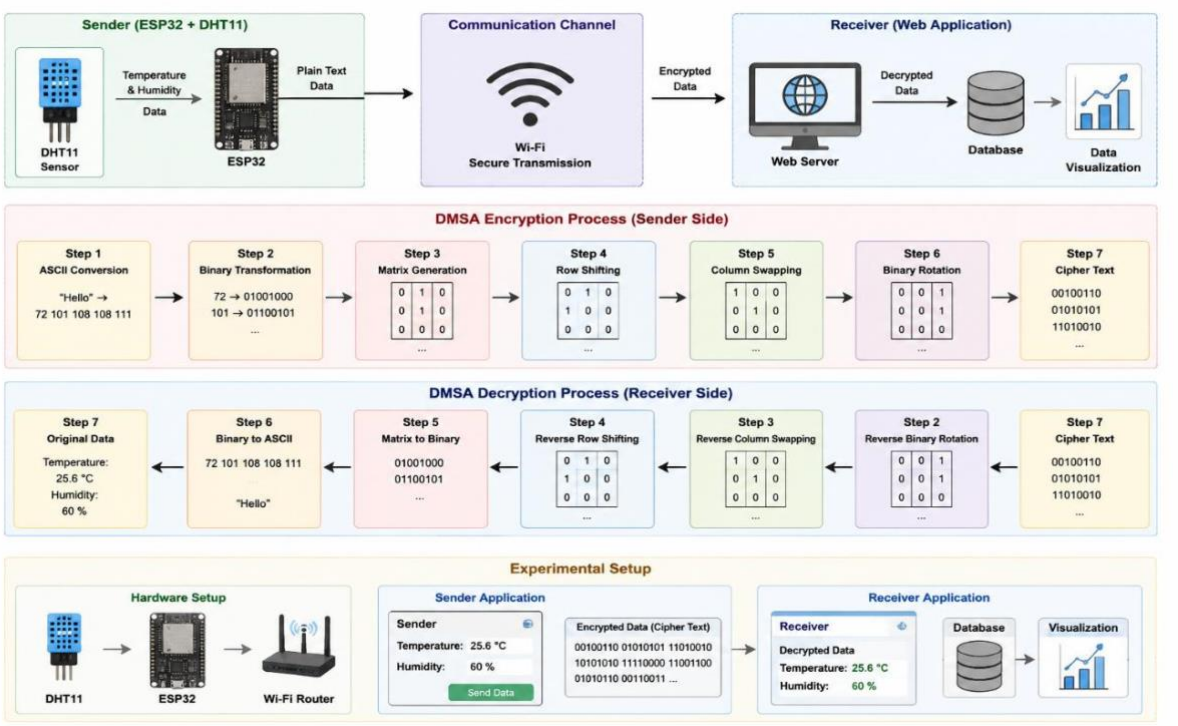


Figure 1. Proposed DMSA-based IoT communication architecture.

3.1. Hardware Components

The hardware section of the proposed system includes ESP32 and DHT11 sensors. ESP32 is a low-power microcontroller with built-in WiFi and Bluetooth capabilities, making it suitable for IoT-based applications. The DHT11 sensor is used for collecting real-time temperature and humidity data.

The sensor continuously measures environmental conditions and sends the collected values to the ESP32 microcontroller through GPIO communication. ESP32 then transfers the sensor data to the sender application through serial communication using the COM port. The hardware components are listed in Table 1, and the hardware connection is shown in Figure 2.

Table 1. Hardware components of the proposed system.

Component	Description
ESP32-WROOM-32	Microcontroller for sensor communication
DHT11 Sensor	Temperature and humidity sensor
Breadboard	Circuit connection platform
Jumper Wires	Hardware connections

Component	Description
USB Cable	ESP32 programming and power

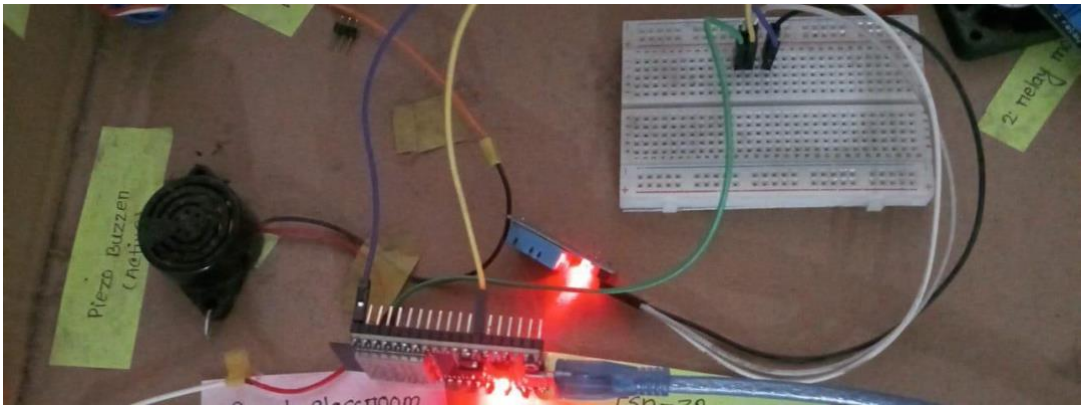


Figure 2. ESP32 and DHT11 sensor hardware connection.

3.2. Software Components

The software implementation of the proposed system was developed using Arduino IDE, Python, Flask framework, and supporting communication libraries. Arduino IDE was used for programming the ESP32 microcontroller and collecting real-time temperature and humidity data from the DHT11 sensor.

The sender and receiver applications were developed using Python and Flask web framework to create web-based communication interfaces. The PySerial library was used to establish serial communication between ESP32 and the sender application. In addition, the Requests library was used to transmit encrypted data from the sender application to the receiver application through HTTP communication.

Visual Studio Code (VS Code) was used as the primary development environment for implementing and testing the software modules of the proposed framework.

3.3. Sender and Receiver Architecture

The proposed communication framework contains two software modules: the sender application and the receiver application. Both applications were developed using Python and Flask web framework.

The sender application receives real-time sensor data from ESP32 and performs DMSA-based encryption (Figure 3). After encryption, the generated cipher text and intermediate transformation results are transmitted to the receiver application. The receiver application performs reverse DMSA operations including reverse rotation, reverse column swapping, and reverse row shifting to reconstruct the original sensor data (Figure 4).

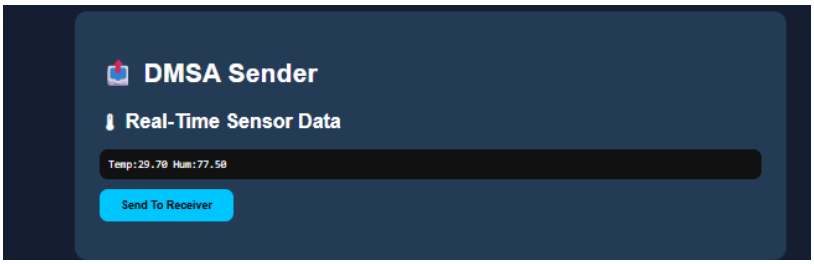


Figure 3. Sender web application with real-time sensor data.

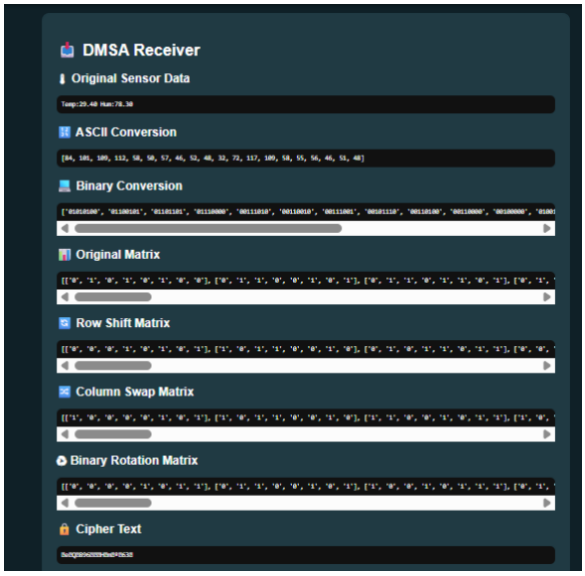


Figure 4. Receiver application showing the encryption and decryption process.

3.4. DMSA Encryption Process

The proposed Dynamic Matrix Shift Algorithm (DMSA) consists of multiple lightweight matrix-based transformation stages designed for secure IoT communication.

Initially, the sensor data is converted into ASCII values. The ASCII values are then transformed into 8-bit binary representations. The generated binary values are organized into matrix structures where row shifting operations are performed using a secret numeric key.

After row shifting, selected matrix columns are swapped to increase randomness and complexity within the binary structure. Binary rotation operations are then applied to further obfuscate the original sensor information. Finally, the transformed binary values are converted into cipher text for secure transmission.

At the receiver side, reverse DMSA operations are executed sequentially to reconstruct the original sensor data from the encrypted cipher text. The complete workflow is shown in Figure 5.

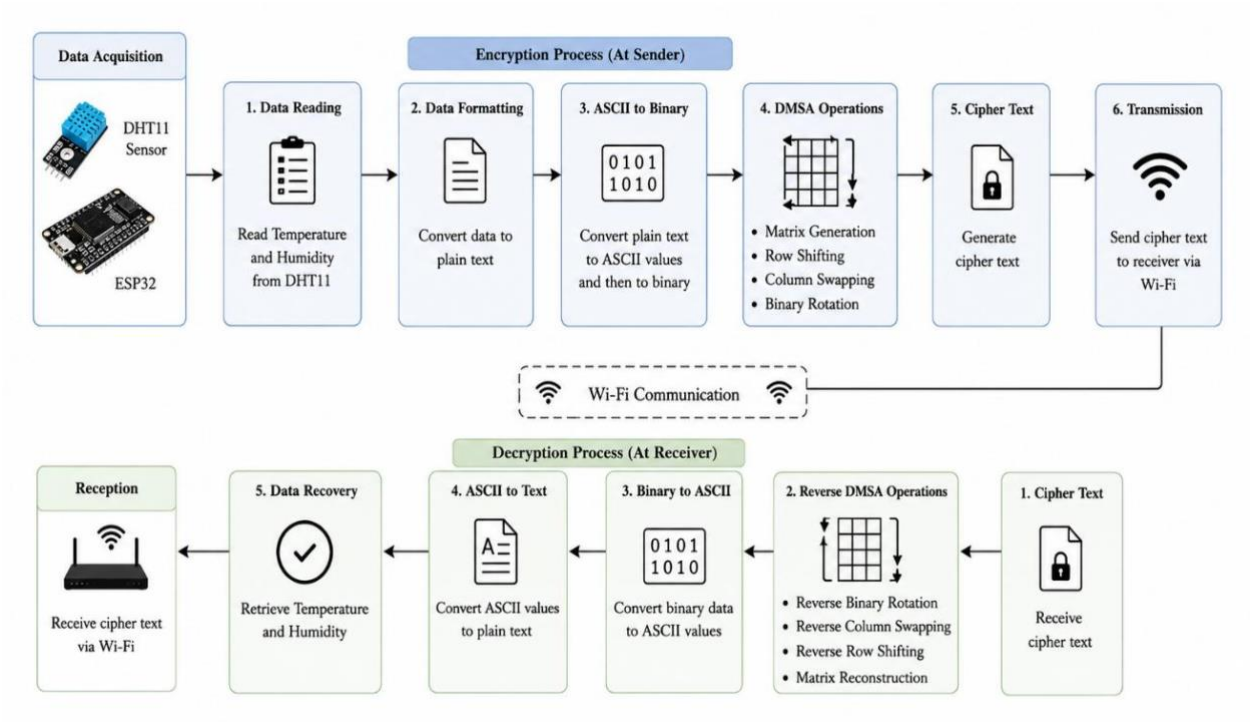


Figure 5. DMSA encryption and decryption workflow.

4. Results

The proposed DMSA-based IoT communication framework was successfully implemented using ESP32, DHT11 sensors, Python Flask applications, and matrix-based encryption operations. The system successfully collected real-time temperature and humidity data from the DHT11 sensor through ESP32 and transmitted the data securely between sender and receiver applications.

The sender application continuously displayed real-time sensor readings collected from ESP32 (Figure 6). After pressing the "Send to Receiver" button, the DMSA algorithm encrypted the sensor data through multiple transformation stages including ASCII conversion, binary transformation, matrix generation, row shifting, column swapping, and binary rotation.



Figure 6. Sender application displaying real-time sensor data.

The encrypted cipher text generated by the DMSA algorithm appeared significantly different from the original sensor information, making the transmitted data difficult to interpret by unauthorized users (Figure 7).



Figure 7. Generated cipher text after DMSA encryption.

The receiver application successfully received the encrypted data and performed reverse DMSA operations to reconstruct the original sensor data. The receiver interface displayed all intermediate transformation stages including ASCII values, binary matrices, shifted matrices, swapped matrices, rotated matrices, cipher text, and decrypted output (Figure 8).

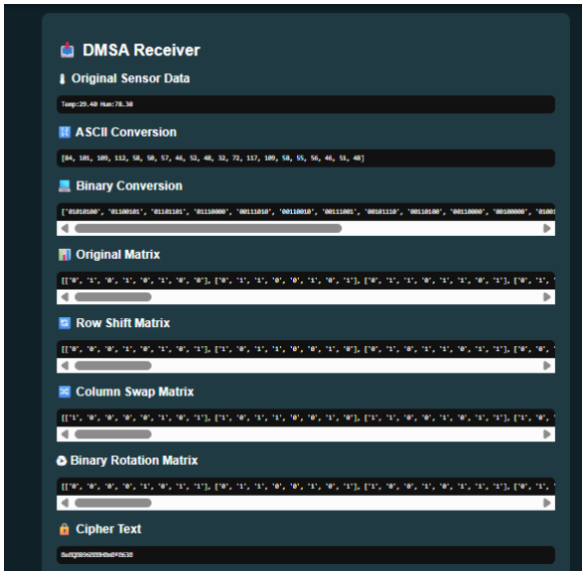


Figure 8. Receiver application showing the full encryption and decryption process.

Experimental observations demonstrated that the proposed lightweight encryption framework successfully maintained secure communication while preserving accurate recovery of the original temperature and humidity data at the receiver side.

Table 2 presents a sample comparison between original sensor data and decrypted output.

Table 2. Comparison between original and decrypted sensor data.

Original Sensor Data	Decrypted Sensor Data
Temp: 29.10 Hum: 77.90	Temp: 29.10 Hum: 77.90
Temp: 30.20 Hum: 85.70	Temp: 30.20 Hum: 85.70

The results indicate that the proposed DMSA framework can successfully secure real-time IoT sensor communication with low computational complexity suitable for lightweight embedded systems.

5. Discussion

The experimental implementation demonstrated that the proposed Dynamic Matrix Shift Algorithm (DMSA) can successfully provide lightweight security for real-time IoT communication systems. The integration of ESP32, DHT11 sensors, and Flask-based sender-receiver applications enabled secure transmission of environmental sensor data with low computational overhead.

The proposed DMSA algorithm applied multiple matrix-based transformations including row shifting, column swapping, and binary rotation to increase the complexity of transmitted data. These operations generated encrypted cipher text significantly different from the original sensor information, reducing the possibility of direct interpretation by unauthorized users.

Compared with traditional encryption techniques such as AES and RSA, the proposed approach requires lower computational resources and simpler processing operations, making it more suitable for lightweight embedded systems and low-power IoT devices. The experimental results confirmed that the receiver application successfully reconstructed the original sensor data through reverse DMSA operations without data loss.

The proposed system also provided real-time visualization of the encryption and decryption process through web-based sender and receiver interfaces. This feature improved the understanding of lightweight cryptographic operations and secure communication workflows in IoT environments.

However, the proposed framework has several limitations. The current implementation focuses mainly on educational and lightweight security purposes rather than industrial-grade cryptographic protection. In addition, the secret key management mechanism is static, which may reduce security strength in large-scale communication systems. Network-level attacks and advanced cryptographic threat models were not evaluated in this study.

Despite these limitations, the proposed research contributes a lightweight and practical secure IoT communication framework suitable for educational purposes, lightweight embedded systems, and real-time sensor monitoring applications.

6. Conclusion and Future Direction

This research presented a lightweight secure IoT communication framework using ESP32, DHT11 sensors, and the proposed Dynamic Matrix Shift Algorithm (DMSA). The developed system successfully collected real-time temperature and humidity data and securely transmitted the information between sender and receiver applications through multiple matrix-based encryption operations.

The proposed DMSA algorithm applied several lightweight transformation stages including ASCII conversion, binary transformation, matrix generation, row shifting, column swapping, and binary rotation to generate encrypted cipher text. The receiver application successfully reconstructed the original sensor data using reverse DMSA operations without data loss.

The experimental implementation demonstrated that the proposed framework can provide secure real-time IoT communication while maintaining low computational complexity suitable for lightweight embedded systems. The proposed system also improved the visualization and understanding of encryption and decryption processes in IoT communication environments.

Future improvements may include dynamic key generation, cloud integration, database connectivity, wireless communication optimization, and hybrid cryptographic implementations for enhanced security and scalability in large-scale IoT applications.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1) M. Rana, Q. Mamun, and R. Islam, “Lightweight cryptography in IoT networks: A survey,” Future Generation Computer Systems, vol. 129, pp. 77–89, 2022, doi: 10.1016/j.future.2021.11.011.
2) Y. Zhong and J. Gu, “Lightweight block ciphers for resource-constrained environments: A comprehensive survey,” Future Generation Computer Systems, vol. 157, 2024, doi: 10.1016/j.future.2024.03.054.
3) I. Radhakrishnan, S. Jadon, and P. B. Honnavalli, “Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices,” Sensors, vol. 24, no. 12, p. 4008, 2024, doi: 10.3390/s24124008.
4) F. Alcaraz-Velasco, J. M. Palomares, and J. Olivares, “GS3: A Lightweight Method of Generating Data Blocks With Shuffling, Scrambling, and Substituting Data for Constrained IoT Devices,” IEEE Internet of Things Journal, vol. 11, no. 15, pp. 25782–25801, 2024, doi: 10.1109/JIOT.2024.3395543.
5) S. A. Ansari and S. Ali, “A systematic review of lightweight cryptographic schemes for security and privacy in IoT,” Discover Computing, vol. 28, art. no. 266, 2025, doi: 10.1007/s10791-025-09755-3.
6) Y. Zhong and J. Gu, “Lightweight block ciphers for resource-constrained environments: A comprehensive survey,” Future Generation Computer Systems, vol. 157, 2024, doi: 10.1016/j.future.2024.03.054.
7) I. Radhakrishnan, S. Jadon, and P. B. Honnavalli, “Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices,” Sensors, vol. 24, no. 12, p. 4008, 2024, doi: 10.3390/s24124008.
8) F. Alcaraz-Velasco, J. M. Palomares, and J. Olivares, “GS3: A Lightweight Method of Generating Data Blocks With Shuffling, Scrambling, and Substituting Data for Constrained IoT Devices,” IEEE Internet of Things Journal, vol. 11, no. 15, pp. 25782–25801, 2024, doi: 10.1109/JIOT.2024.3395543.
9) S. A. Ansari and S. Ali, “A systematic review of lightweight cryptographic schemes for security and privacy in IoT,” Discover Computing, vol. 28, art. no. 266, 2025, doi: 10.1007/s10791-025-09755-3.
10) B. Gușiță et al., “Securing IoT edge: a survey on lightweight cryptography, anonymous routing and communication protocol enhancements,” International Journal of Information Security, vol. 24, art. no. 149, 2025, doi: 10.1007/s10207-025-01071-7.

Publisher’s Note: Noesis Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.