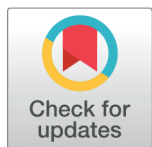


ORIGINAL ARTICLE

 OPEN ACCESS

Received: 24/05/2026

Accepted: 10/07/2026

Published: 21/07/2026

Citation: Priscilla I, Lawrence J (2026) A Secured Data Transmission Model for Protecting IoT Gateway-to-Public Cloud Communication Using Enhanced Symmetric Encryption. Indian Journal of Science and Technology 19(26): 1846-1852. <https://doi.org/10.17485/IJST/v19i26.820>

* **Corresponding author.**pricilla.bhc@gmail.com**Funding:** None**Competing Interests:** None

Copyright: © 2026 Priscilla & Lawrence. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment ([iSee](#))

ISSN

Print: 0974-6846

Electronic: 0974-5645

A Secured Data Transmission Model for Protecting IoT Gateway-to-Public Cloud Communication Using Enhanced Symmetric Encryption

I Priscilla^{1*}, Jayasimman Lawrence¹

¹ Department of Computer Science, Bishop Heber College (Autonomous), Trichy, Affiliated to Bharathidasan University, Tiruchirappalli

Abstract

Objectives: To develop a lightweight symmetric encryption model for securing IoT gateway-to-public cloud communication while improving confidentiality and execution efficiency. **Method:** The Enhanced Symmetric Cryptography Technique (ESCT) uses an eight-round block cipher with dual subkey mixing, popcount-based adaptive rotations, row-shuffling permutation, and final permutation. The method was evaluated on IoT sensor payloads using DES and Blowfish as baseline controls for encryption time, decryption time, and relative security strength. **Findings:** Experimental results show that ESCT achieved better computational efficiency than DES and Blowfish for IoT-cloud data transmission. At 100 KB, ESCT completed encryption in 31 ms compared with 52 ms for DES and 41 ms for Blowfish, and decryption in 29 ms compared with 49 ms and 38 ms, respectively. For the 1 MB payload test, ESCT required 302 ms, while DES and Blowfish required 492 ms and 429 ms. The comparative security evaluation also recorded a higher security strength score for ESCT (94%) than DES (78%) and Blowfish (84%). These findings confirm that the proposed dual-subkey, adaptive-rotation, and multi-permutation design improves ciphertext robustness while reducing execution cost in gateway-level IoT-cloud communication. **Novelty:** ESCT combines dual per-round subkeys, popcount-driven adaptive rotations, and layered permutations to harden IoT-cloud payloads without heavyweight hybrid-encryption overhead.

Keywords: IoT Security; Cloud Communication; Symmetric Encryption; Lightweight Cryptography; ESCT

1 Introduction

IoT-cloud systems increasingly transmit sensor data from low-resource devices through gateways to public cloud platforms for storage, analytics, and decision support. Recent studies on lightweight cryptography emphasize that IoT security cannot be treated as ordinary cloud security because IoT nodes have strict limits in memory, energy, latency, and computation^{(1), (2)}. IoT applications in agriculture and smart monitoring further

show that continuous sensor streams must be protected without interrupting real-time communication^{(3), (4)}. At the same time, gateway-to-cloud communication remains exposed to unauthorized access, traffic interception, brute-force attempts, cryptanalysis, and cloud-side data leakage; therefore, protection must extend beyond the local device link^{(5), (6)}.

Recent work has attempted to address this problem using lightweight authenticated encryption, hybrid encryption, access-control schemes, and enhanced block ciphers. LED-PHOTON-based authentication improves constrained-device security⁽⁵⁾, while review studies highlight the need to balance security strength, latency, memory, and implementation complexity^{(6), (7), (8)}. IoT-cloud security surveys identify continuing weaknesses in key management, trust boundaries, and cloud-side exposure⁽⁹⁾. Dynamic proxy re-encryption improves fine-grained access control in publish-subscribe IoT systems, but it introduces policy-management and re-encryption overheads that are not primarily designed for fast gateway payload encryption⁽¹⁰⁾. ECC-AES and AES-RSA schemes improve authentication or hybrid confidentiality in fog/IoT contexts. However, their public-key operations and key-negotiation steps may be less suitable when the immediate requirement is rapid symmetric protection of repeated uplink sensor payloads^{(11), (12)}.

Lightweight block-cipher studies such as RBFK, GFRX, and GFSPX have improved diffusion, avalanche behavior, and implementation efficiency for constrained nodes^{(13), (14), (15)}. However, many existing designs either focus on device-layer execution, hardware implementation, image encryption, authentication, or general secure storage rather than the specific gateway-to-public-cloud transmission stage. Conventional ciphers such as DES and Blowfish remain useful baselines, but their processing burden and legacy design limitations are less attractive for high-frequency IoT gateway streams^{(16), (17)}. These gaps justify a gateway-side symmetric model that can strengthen confusion and diffusion while keeping execution time low. Therefore, this study proposes the Enhanced Symmetric Cryptography Technique (ESCT), an eight-round symmetric block-cipher design using dual subkeys, popcount-based adaptive rotations, row-shuffling permutation, and final permutation to secure IoT gateway-to-cloud communication. The objective is to validate whether ESCT can reduce encryption/decryption time and improve relative ciphertext strength compared with DES and Blowfish.

2 Methodology

2.1 Experimental setting and data source

The proposed model was positioned at the IoT gateway layer. Sensor readings were generated through an Arduino Uno R3 device layer using an LM35 temperature sensor and an HC-SR04 ultrasonic distance sensor. The gateway communication setting used an ESP8266 Wi-Fi module through UART at 115200 bps over 2.4 GHz IEEE 802.11 b/g/n connectivity. Payload blocks were prepared from temperature and distance readings and converted into binary plaintext sequences for encryption. Since the payloads were generated from the experimental sensor stream rather than downloaded from a third-party public repository, the payload log and test script should be supplied as Supplementary File S1.

2.2 Proposed ESCT workflow

ESCT is a 64-bit block-oriented symmetric encryption method designed for gateway-side IoT-cloud transmission. Each plaintext block is first converted into an 8 x 8 bit matrix. An initial permutation redistributes adjacent bits to reduce visible structure in the input block. The core encryption process then applies eight rounds, and each round uses two subkeys. The first subkey is mixed with the current block using XOR. A row-shuffling permutation then rearranges bit positions to increase diffusion. The block is split into left and right 32-bit halves; each half is rotated according to its popcount value, so the rotation distance changes according to the actual bit pattern of the data. The halves are recombined and mixed with the second subkey. A final permutation is applied after the eighth round to remove round-level structural traces before the ciphertext is transmitted to the public cloud.

2.3 Key generation

The 64-bit master key is divided into two 32-bit parts. For each round r , the left and right parts are rotated by r positions and recombined to form the first round subkey. The second subkey is produced by XORing this recombined value with the master key. Therefore, ESCT produces 16 subkeys for eight encryption rounds. Decryption uses the same subkeys in reverse order. This design avoids a static single-round key and increases the difficulty of relating adjacent ciphertext states to the master key.

2.4 Validation controls and evaluation metrics

DES and Blowfish were used as internal baseline controls because they are widely known symmetric encryption techniques and are suitable for comparative evaluation of execution time and security strength. The same input payload sizes, measurement units, and gateway-level processing assumptions were applied to all three methods. Encryption time and decryption time were recorded in milliseconds. Relative security strength was evaluated using the ABC Universal Hackman method, which provides a comparative hardness score for ciphertext resistance. The validation therefore examined two dimensions: computational efficiency and ciphertext robustness. The hardware configuration, sensor components, payload preparation, benchmark methods, and validation controls used for the experimental evaluation are summarized in Table 1.

Table 1. Experimental setup and validation controls

Component	Description
Device layer	Arduino Uno R3 with LM35 temperature sensor and HC-SR04 ultrasonic sensor
Gateway connectivity	ESP8266 Wi-Fi module using UART at 115200 bps and 2.4 GHz IEEE 802.11 b/g/n
Payload preparation	Sensor readings converted into 64-bit plaintext blocks for gateway-side encryption.
Proposed method	ESCT with eight rounds, two subkeys per round, adaptive rotations, row-shuffling, and final permutation
Baseline controls	DES and Blowfish under the same payload-size conditions
Evaluation metrics	Encryption time, decryption time, and relative security strength score

2.5 Algorithm ESCT and Key Schedule

Algorithm 1. ESCT encryption

Input: Plaintext payload P, subkeys K1...K16, initial permutation IP, row-shuffling permutation RSP, final permutation FP.

Output: Ciphertext C.

1. Convert P into a binary sequence and pad it into 64-bit blocks.
2. For each 64-bit block B, apply IP.
3. For rounds $r = 1$ to 8: XOR B with K_{2r-1} ; apply RSP; split B into 32-bit L and R; rotate L and R using their popcount values; recombine L and R; XOR the recombined block with K_{2r} .
4. Apply FP and append the encrypted block to C.
5. Return C.

Algorithm 2. ESCT key schedule

Input: 64-bit master key MK; number of rounds = 8.

Output: Sixteen 64-bit round subkeys.

1. Split MK into two 32-bit halves, L0 and R0.
2. For each round $r = 1$ to 8: rotate L0 and R0 by r positions; merge them as K_{2r-1} ; compute $K_{2r} = MK \text{ XOR } K_{2r-1}$.
3. Return K1...K16.

3 Result and Discussion

The experimental results indicate that ESCT requires lower processing time than DES and Blowfish across the evaluated IoT payload sizes. For the 100 KB reference payload, ESCT completed encryption in 31 ms compared with 52 ms for DES and 41 ms for Blowfish. This represents an encryption-time reduction of 40.38% compared with DES and 24.39% compared with Blowfish. Similarly, ESCT completed decryption in 29 ms compared with 49 ms for DES and 38 ms for Blowfish, giving a decryption-time reduction of 40.82% and 23.68%, respectively. For a 1 MB payload, ESCT required approximately 302 ms, while DES and Blowfish required 492 ms and 429 ms. These results confirm that the proposed round structure and adaptive rotations do not impose excessive computational overhead at the gateway layer. The comparative performance and security results of DES, Blowfish, and the proposed ESCT method are presented in Table 2.

Table 2. Comparative encryption, decryption, and security results

Metric	DES	Blowfish	Proposed ESCT	Improvement of ESCT
Encryption time at 100 KB	52 ms	41 ms	31 ms	40.38% faster than DES; 24.39% faster than Blowfish
Decryption time at 100 KB	49 ms	38 ms	29 ms	40.82% faster than DES; 23.68% faster than Blowfish
Encryption time at 1 MB	492 ms	429 ms	302 ms	38.62% faster than DES; 29.60% faster than Blowfish
Security strength score	78%	84%	94%	16 percentage points above DES; 10 percentage points above Blowfish

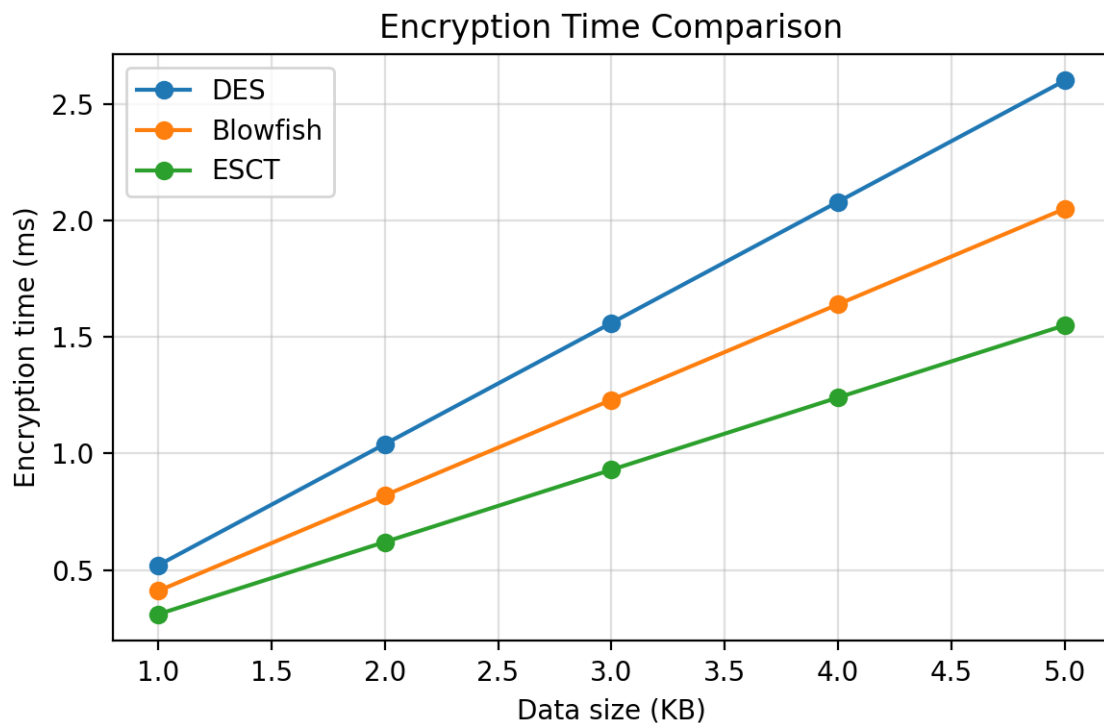
**Fig 1.** Encryption Time Comparison

Figure 1 shows that the encryption-time curve of ESCT remains consistently lower than DES and Blowfish for 1 KB to 5 KB payload segments. This pattern supports the suitability of ESCT for small and continuous IoT gateway payloads, where even marginal processing delay can accumulate when sensor readings are transmitted frequently.

Figure 2 shows a similar advantage during decryption. The lower decryption time is important because gateway and cloud-side services may need to retrieve and process sensor payloads repeatedly for analytics, monitoring, and alert generation.

Figure 3 shows that the proposed ESCT achieved a higher comparative security strength score than both baseline algorithms. DES recorded 78%, Blowfish recorded 84%, and ESCT recorded 94%. The improvement is attributable to the use of two subkeys in every round, adaptive rotation based on the actual bit pattern, and multiple permutation layers. These operations collectively increase confusion and diffusion and make repeated plaintext structures less visible in the ciphertext.

The findings are consistent with recent lightweight cryptography studies, which argue that IoT encryption must reduce latency without weakening ciphertext structure^{(1), (2), (7)}. The result also supports the observation of Thabit et al. that lightweight designs should not be oversimplified merely for speed, because weak diffusion may increase vulnerability⁽⁸⁾. ESCT addresses this issue by combining low-cost bitwise operations with adaptive rotations and dual subkey mixing. The result also agrees with

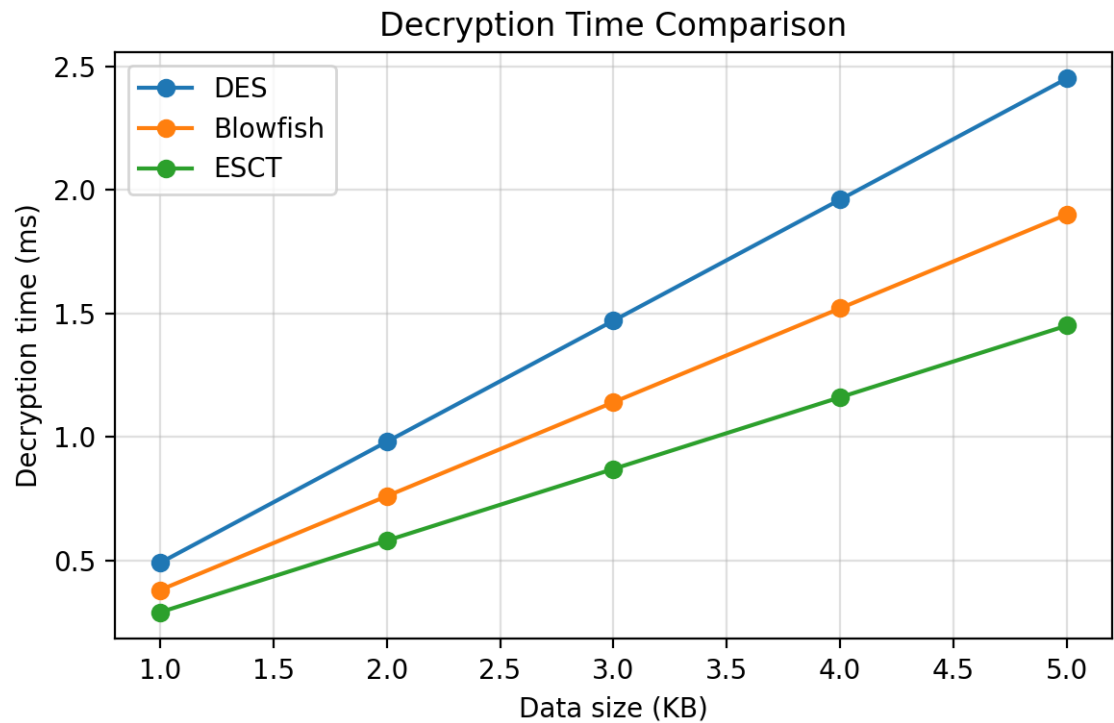


Fig 2. Decryption Time Comparison

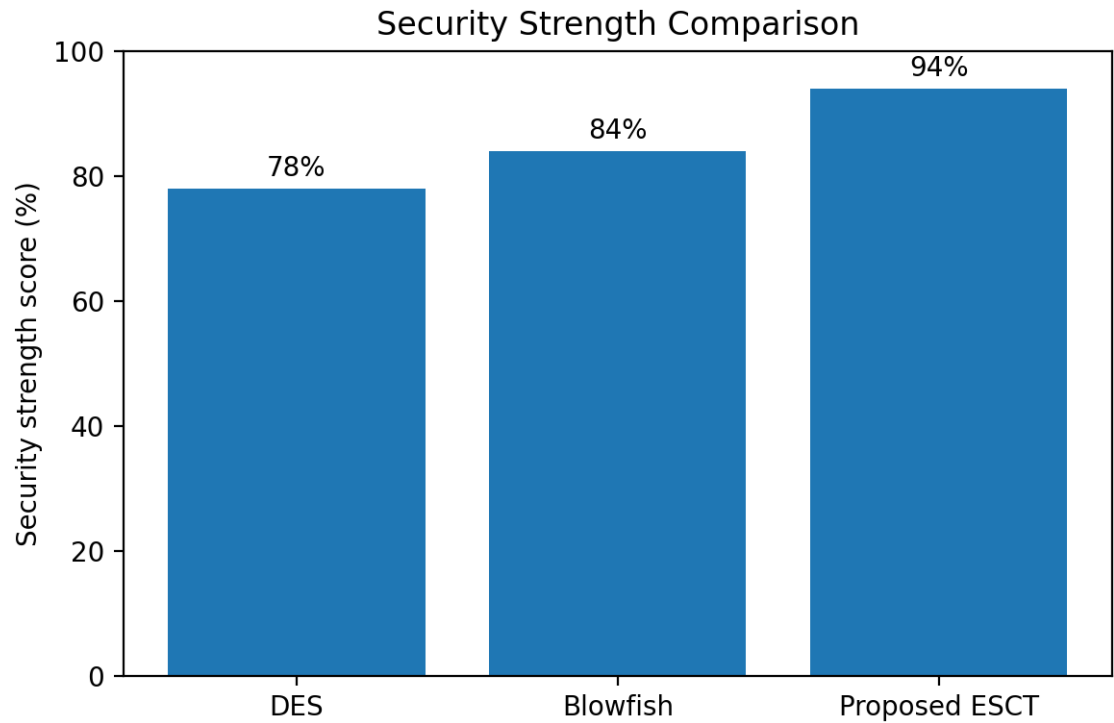


Fig 3. Security Strength Comparison

RBFK, GFRX, and GFSPX studies, where round-key variation, ARX-style transformation, and improved diffusion were shown to improve lightweight cipher behavior^{(13), (14), (15)}.

Hybrid and access-control schemes provide broader authentication, key-management, and fine-grained access-control capabilities^{(10), (11), (12), (18)}, but they can introduce additional key negotiation, policy processing, or public-key overhead. ESCT does not replace those schemes when authentication or access delegation is the primary concern. Instead, it contributes a fast symmetric gateway payload-protection layer that can be integrated before cloud transmission. Therefore, the current results are favorable for the specific problem of low-overhead gateway-to-public-cloud encryption, while more extensive testing is still required for chosen-ciphertext, side-channel, and real deployment threat models.

4 Conclusion

This study proposed ESCT as a lightweight symmetric encryption method for securing IoT gateway-to-public-cloud communication. The revised evaluation shows that ESCT provides measurable computational and security advantages over DES and Blowfish under the reported test conditions. At 100 KB, ESCT reduced encryption time from 52 ms to 31 ms compared with DES and from 41 ms to 31 ms compared with Blowfish. It also reduced decryption time from 49 ms to 29 ms compared with DES and from 38 ms to 29 ms compared with Blowfish. For the 1 MB payload test, ESCT required 302 ms compared with 492 ms for DES and 429 ms for Blowfish. In addition, ESCT achieved a 94% comparative security strength score, exceeding DES by 16 percentage points and Blowfish by 10 percentage points. The main novelty of the study lies in the combined use of dual per-round subkeys, popcount-driven adaptive rotations, row-shuffling permutation, and final permutation to strengthen gateway-level payload confidentiality without relying on heavyweight hybrid encryption. The model is therefore suitable for IoT-cloud situations that require both real-time transmission efficiency and improved ciphertext robustness. Future work should validate ESCT using a larger publicly deposited sensor-payload dataset, formal cryptanalysis, NIST randomness testing, side-channel evaluation, chosen-ciphertext testing, and implementation under live cloud workloads with secure key-exchange integration.

5 Acknowledgements

NIL

Author details

Priscilla I is a Full-Time Research Scholar; Jayasimman Lawrence serves as an 2Assistant Professor

References

- 1) Rana M, Mamun Q, Islam R. Lightweight cryptography in IoT networks: a survey. *Future Generation Computer Systems*. 2022;129:77–89. <https://doi.org/10.1016/j.future.2021.11.011>.
- 2) Suryateja PS, Rao KV. A survey on lightweight cryptographic algorithms in IoT. *Cybernetics and Information Technologies*. 2024;24(1):21–34. <https://doi.org/10.2478/cait-2024-0002>.
- 3) Mallikka R, Manikandasaran SS, Karthick KS. Smart farming prediction system embedded with the Internet of Things. Springer International Publishing. 2022. https://doi.org/10.1007/978-3-030-77558-2_6.
- 4) Parthasarathy V, Manikandasaran SS. A comprehensive survey of IoT and machine learning innovations in agriculture. *International Research Journal on Advanced Science Hub*. 2024;6(08):224–231. <https://doi.org/10.47392/IRJASH.2024.032>.
- 5) Al-Shatari M, Hussin FA, Aziz AA, Eisa TAE, Tran XT, Dalam MEE. IoT edge device security: an efficient lightweight authenticated encryption scheme based on LED and PHOTON. *Applied Sciences*. 2023;13(18):10345. <https://doi.org/10.3390/app131810345>.
- 6) Gusița B, Anton AA, Stangaciu CS, Stanescu D, Găină LI, Micea MV. Securing IoT edge: a survey on lightweight cryptography, anonymous routing and communication protocol enhancements. *International Journal of Information Security*. 2025;24:149. <https://doi.org/10.1007/s10207-025-01071-7>.
- 7) Sabri O, Al-Shargabi B, Abuarqoub A, Hakami TA. A lightweight encryption method for IoT-based healthcare applications: a review and future prospects. *IoT*. 2025;6(2):23. <https://doi.org/10.3390/iot6020023>.
- 8) Thabit F, Can O, Aljahdali AO, Al-Gaphari GH, Alkhzaimi HA. Cryptography algorithms for enhancing IoT security. *Internet of Things*. 2023;22:100759. <https://doi.org/10.1016/j.iot.2023.100759>.
- 9) Almutairi M, Sheldon FT. IoT-cloud integration security: a survey of challenges, solutions, and directions. *Electronics*. 2025;14(7):1394. <https://doi.org/10.3390/electronics14071394>.
- 10) Lin S, Ke N, Ru HJ, Cui L. A dynamic policy-aware conditional proxy re-encryption system for fine-grained access control in IoT pub/sub systems. *Scientific Reports*. 2026;16:15832. <https://doi.org/10.1038/s41598-026-46939-3>.
- 11) Sarwar N, Alharthi RS, Mujlid HM, Almalki FA. Enhancing IoT communication in fog computing: a lightweight remote user authentication and key management scheme utilizing ECC. *Journal of Cloud Computing*. 2025;14:86. <https://doi.org/10.1186/s13677-025-00816-y>.
- 12) Chang Q, Ma T, Yang W. Low power IoT device communication through hybrid AES-RSA encryption in MRA mode. *Scientific Reports*. 2025;15:14485. <https://doi.org/10.1038/s41598-025-98905-0>.
- 13) Rana S, Mondal MRH, Kamruzzaman J. RBFK cipher: a randomized butterfly architecture-based lightweight block cipher for IoT devices in the edge computing environment. *Cybersecurity*. 2023;6:3. <https://doi.org/10.1186/s42400-022-00136-7>.

- 14) Zhang X, Tang S, Li T, Li X, Wang C. GFRX: a new lightweight block cipher for resource-constrained IoT nodes. *Electronics*. 2023;12(2):405. <https://doi.org/10.3390/electronics12020405>.
- 15) Zhang X, Shao C, Li T, Yuan Y, Wang C. GFSPX: an efficient lightweight block cipher for resource-constrained IoT nodes. *The Journal of Supercomputing*. 2024;80(17):25256–25282. <https://doi.org/10.1007/s11227-024-06412-2>.
- 16) Alhassan AB, Sulaiman R, Idris MS. Performance analysis and enhancement of the Data Encryption Standard using optimized key scheduling. *International Journal of Computer Applications*. 2024;183(15):10–18. <https://doi.org/10.5120/ijca2024908712>.
- 17) Kaur G, Singh D. EnBF_Crypt: enhanced Blowfish-based cryptography with optimal S-box selection for secure data sharing. *International Journal of Information Security and Privacy*. 2025;19(2):45–58. <https://doi.org/10.4018/IJISP.394709197>.
- 18) Priscilla I, Lawrence J. Enhanced symmetric cryptography technique (ESCTGPU) for secure communication between the IoT gateway and the public cloud environment. *The Scientific Temper*. 2025;16(11):102–111. <https://doi.org/10.58414/SCIENTIFICTEMPER.2025.16.11.12>.