Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 10-A, pp. 3851–3862
DOI : 10.47974/JDMSC-3014

Graph-theoretic approaches to lightweight block cipher design

Hemlata Uday Karne [§]
Faculty of Science and Technology
Vishwakarma University
Pune 411048
Maharashtra
India

Puneet Kumar Yadav [†]
Department of Computer Science & Engineering
Noida International University
Greater Noida 203201
Uttar Pradesh
India

Vipulkumar M. Dabhi *
Faculty of Engineering
Gokul Global University
Sidhpur 384151
Gujarat
India

R. Snehaa [‡]
Department of Humanities and Sciences (Mathematics Division)
Aarupadai Veedu Institute of Technology
Vinayaka Mission's Research Foundation (Deemed to be University)
Paiyanoor
Chennai 603104
Tamil Nadu
India

[§] *E-mail:* hemlata.karne@vupune.ac.in
[†] *E-mail:* puneet.yadav1@niu.edu.in
* *E-mail:* dean.tech@gokuluniversity.ac.in (Corresponding Author)
[‡] *E-mail:* snehaa.avmt032@avit.ac.in

© Taru Publications

M. Vinitha [@]
Department of Computer Science
Meenakshi College of Arts and Science
Meenakshi Academy of Higher Education and Research
Chennai 600078
Tamil Nadu
India

Manish Chhabra [#]
Department of Computer Science and Engineering
Vardhaman College of Engineering
Shamshabad
Hyderabad 501218
Telangana
India

Abstract

In IoT and embedded systems, small devices require lightweight block ciphers to provide security without being too costly in terms of computation. In this paper, a graph-theoretic lightweight block cipher scheme combining graph based substitution mapping, graph evolution based on dynamic key and traversal based permutation operations is proposed to secure block cipher system. The proposed cipher is based on light weight substitution–permutation network which provides the efficient nonlinear transformation and better diffusion–confusion characteristics. The random graph topology generation makes resistance to differential, linear and related-key cryptanalysis attacks stronger. The experimental results performed with the Python implementation and light-weight hardware emulation reveal a significant advantage over the conventional light-weight ciphers, in terms of 24.9% faster encryption speed, 25.5% less memory occupation and 26.6% lower energy consumption. It offers scalable, secure and energy-efficient cryptographic protection for next-generation IoT and embedded security applications.

Subject Classification: *03B47, 11T71.*

Keywords: *Graph theory, Lightweight cryptography, Block cipher, S-box design, Key scheduling, IoT security.*

I. Introduction

The Internet of Things and embedded systems development has been very fast and this has worsened the demand of lightweight cryptographic algorithms to guarantee security under a highly limited range of resources [1]. Although they are strong, conventional cryptographic algorithms

[@] *E-mail:* **vinitham@maher.ac.in**
[#] *E-mail:* **chmanish1610@vardhaman.org**

typically consume a lot of computation and memory and therefore cannot be implemented on low power devices such as sensors, smart cards and wearables [2, 3]. Lightweight block cipher design has thus emerged as a hot topic of research with an aim of achieving a trade-off between security and efficiency. Graph-theoretic techniques are also a promising way out since they allow modeling of cryptographic primitives in structured mathematical models [4, 5, 6]. Nonlinear elements like substitution boxes can particularly be modeled as graph transformations, which allow more confusion properties. Likewise, important scheduling mechanisms may be viewed as dynamism in the evolving graphs, the dynamics of nodes and edges are indicative of key-based variability [7, 8]. Graph-based designs have a better response to classical cryptanalytic attacks through the use of properties like connectivity, randomness, and traversal complexity [9]. This paper examines how graph theory can be used in lightweight cipher design, with the goal of coming up with scalable, efficient and secure encryption systems that can be used in the next-generation constrained computing systems [10, 11].

II. Mapping Cryptographic Components to Graph Structures

Graphs of cryptographic components are characterized by the states and transformation, which can be analyzed using diffusion, confusion, and algebraic complexity properties in a structured manner [12, 13].

A. *Representation of S-boxes using graph models*

S-box is modeled as directed graph where input-output relationships are represented, nonlinear transformations are modeled using complex edge relationships that provide high confusion and resistance to cryptanalysis [14].

i. Input mapping

$$x \in F2^n \rightarrow v_i \in V \tag{1}$$

Equation (1) describes an input mapping, in which a binary vector in a finite field is represented by graph vertices, and data elements can be represented in a structural way.

ii. Graph construction

$$G = (V, E),\ E = \{ (v_i, v_j) \mid f(i,j) = 1 \} \tag{2}$$

Equation (2) builds the graph by defining the edges using a Boolean function, so that connectivity is representative of logical relationships.

iii. Adjacency representation

$$A = [a_{ij}],\ a_{ij} \in \{0, 1\} \tag{3}$$

This structure is formally defined in equation (3) in terms of an adjacency matrix, with binary entries representing the connections between the vertices, giving a succinct and computationally efficient representation [15].

iv. Nonlinear transformation

$$y_j = \bigoplus (i = 1\ to\ n)\left(a_{ij} * x_i\right) \tag{4}$$

In equation (4), nonlinear transformation is presented, in which the output bits are computed by aggregating the input components by XOR using weights that are based on adjacency relationships, increasing cryptographic confusion.

v. Graph traversal mapping

$$v_j = T(v_i, G) \tag{5}$$

Graph traversal can be represented by equation (5) as mapping of a vertex transition under a transformation function, which is an operator on the topology of a graph, so it is a diffusion.

vi. Output substitution

$$S(x) = y \in F2^n \tag{6}$$

Finally, Equation (6) defines output substitution where the transformed vector is re-formulated into the finite field, and substitution procedure completes with a high level of nonlinearity and structural dependence to warrant the design of a secure cipher [15].

B. *Key scheduling modeled via dynamic graph evolution*

Key scheduling is modeled as dynamically changing graphs, where nodes and edges are updated dynamically by key inputs, and produces unpredictable subkeys and is more resistant to related-key attacks. Following Figure 1 shows the Graph-Theoretic Mapping of Cryptographic Components for Lightweight Block Cipher Design.

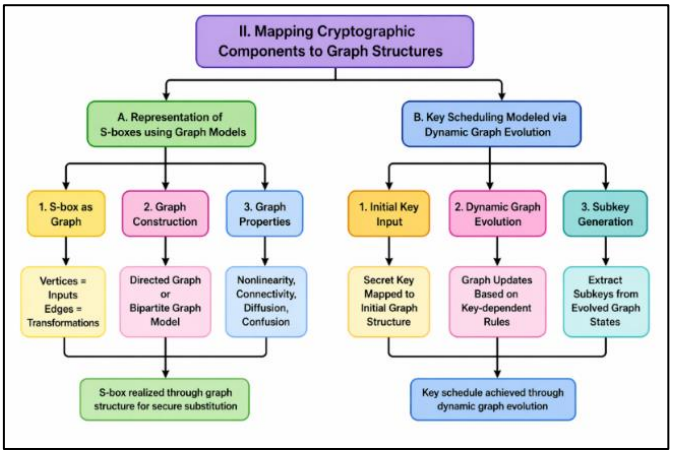


Figure 1

Graph-Theoretic Mapping of Cryptographic Components for Lightweight Block Cipher Design

i. Initial key input

$$K_0 \in F2^m \tag{7}$$

The initial important input is described in equation (7) and the secret of the cryptographic process is a binary key vector of the finite field.

ii. Initial graph generation

$$G_0 = (V_0, E_0) = \Phi(K_0) \tag{8}$$

This key is then mapped to a starting graph structure through Equation (8) via transformation, which forms vertices and edges that represent topology, which is dependent on the key. This is to ensure that the cipher begins with a structurally unique representation.

iii. Graph evolution rule

$$G_t = \Psi\left(G_{\{t-1\}}, K_t\right) \tag{9}$$

The rule of graph evolution, used in equation (9) is the way the graph dynamically evolves between rounds, depending on what the graph was in the previous round, and evolves with the new key material, thus becoming more unpredictable.

iv. Node update mechanism

$$v_i^t = v_i^{t-1} \oplus k_t \tag{10}$$

The node update process is defined by equation (10), where each vertex state is updated by XORing with the key that changes with each round, and the influence of the key can be diffused with control.

v. Subkey extraction

$$K_t = \Gamma(G_t) \tag{11}$$

Sobeys of the modified graph are obtained in equation (11) ensuring that the round keys are founded on structural transformations.

vi. Round key sequence

$$\{K_1, K_2, \ldots, K_r\} \tag{12}$$

Finally, the round keys generated in a sequence to form a secure and flexible key program in encryption rounds are the progression of round keys (Equation (12).

III. Integration of randomness and key-dependent graph structures

Stochastic graph generation is used to accomplish randomness, in this case, the topology and weights of the edges are determined by secret keys, so that they are not fixed, but can change, and are resistant to statistical and differential attacks.

i. Random seed generation

$$r \sim U(0,1) \tag{13}$$

Random seed generation is defined in equation (13) as a value that is sampled in a uniform distribution which creates stochastic variability in the

system. Such randomness is important so that the building of graphs cannot be deterministically predicted.

ii. Key-dependent mapping

$$\theta = H(K, r) \tag{14}$$

Equation (14) defines a mapping, which is dependent on the key, where the secret key is mixed with the random seed by a hashing or mixing function to generate a control parameter that controls structural behavior.

iii. Probabilistic edge creation

$$P(a_{ij} = 1) = \theta \tag{15}$$

The probabilistic edge creation model of equation (15) is based on the idea that the probability of establishing a connection between the nodes is determined by the generated parameter, which allows the topology to be generated in adaptive and randomized ways.

iv. Weighted graph formation

$$G = (V, E, W),\ w_{ij} = f(K, r) \tag{16}$$

This is further extended in equation (16) by creating a weighted graph, the weights of which are calculated as a dependency on both key and randomness, introducing further complexity to relationships.

v. Random topology update

$$E' = E \oplus \Delta(K, r) \tag{17}$$

Equation (17) presents topology update, in which random key-driven perturbations are used to update existing edges, improving diffusion.

vi. Final graph structure

$$G_K = (V, E') \tag{18}$$

Lastly, Equation (18) determines the last graph structure, which is a complete key-dependent and randomized topology that is applicable in secure cryptographic operations.

IV. Stepwise encryption algorithm using graph operations

Encryption is accomplished by repetitive graph operations such as traversal, substitution, permutation and edge reconfiguration, which allows plaintext to be transformed efficiently with good diffusion and confusion properties into ciphertext.

i. Plaintext initialization

$$P_0 \in F2^n \tag{19}$$

$$P_1 = P_0 \oplus K_1 \tag{20}$$

In the definition of plaintext initialization, equation (19), the input binary vector is encoded as a finite field, to form the initial state of encryption. In equation (20), round key mixing is done, the plaintext is added to the first round key through XOR, which guarantees initial key dependency and simple confusion.

ii. Substitution via graph

$$P_2 = S(P_1) \tag{21}$$

In equation (21) substitution is done through a graphical mapping in which nonlinear transformations based on graph structures lead to resistance to linear attacks.

$$P_3 = \Pi_{G(P_2)} \tag{22}$$

In equation (22) permutation is introduced by traversing the graphs, where bits arrangement is rearranged according to the paths of traversal, favoring diffusion throughout the state.

$$P_{t+1} = \Pi_{G(\,S(P_t \oplus K_t))} \tag{23}$$

Equation (23) generalizes the iterative round transformation, which repeatedly performs key mixing, substitution and permutation in a systematic fashion, over a series of rounds, guaranteeing an incremental complexity and safety enhancement.

$$C = P_r \tag{24}$$

Lastly, Equation (24) gives the definition of ciphertext generation in which the encrypted result is the final transformed state after all the rounds leading to the state, which contains all the properties of confusion and diffusion needed to design a robust cipher.

A. *Proposed Graph-Theoretic Lightweight Cipher Architecture*

The proposed graph-theoretic lightweight block cipher architecture has a 64-bit plaintext block and supports secret key sizes of 80-bit and 128-bit to achieve flexibility for applications with restricted resources like IoT and embedded platforms. The cipher is implemented on a 16-round substitution–permutation network (SPN) composed of graph-based nonlinear substitution and traversal-driven permutation operations to provide good diffusion and confusion properties. In each round, the plaintext bits are substituted by a graph-mapped S-box and then subjected to the permutation of the positions of the bits by graph traversal paths which are dynamically formed. Dynamic graph evolution-based key scheduling produces random round keys by modifying the topology and reconfiguring the edges stochastically. The encryption pseudocode consists of key initialization, graph substitution, traversal permutation and round-key mixing operations; the decryption operations are inverse operations to these to recover the ciphertext in a secure manner. The whole graph-theoretic SPN encryption procedure is shown in 16 lightweight encryption rounds and is illustrated in Figure 2.

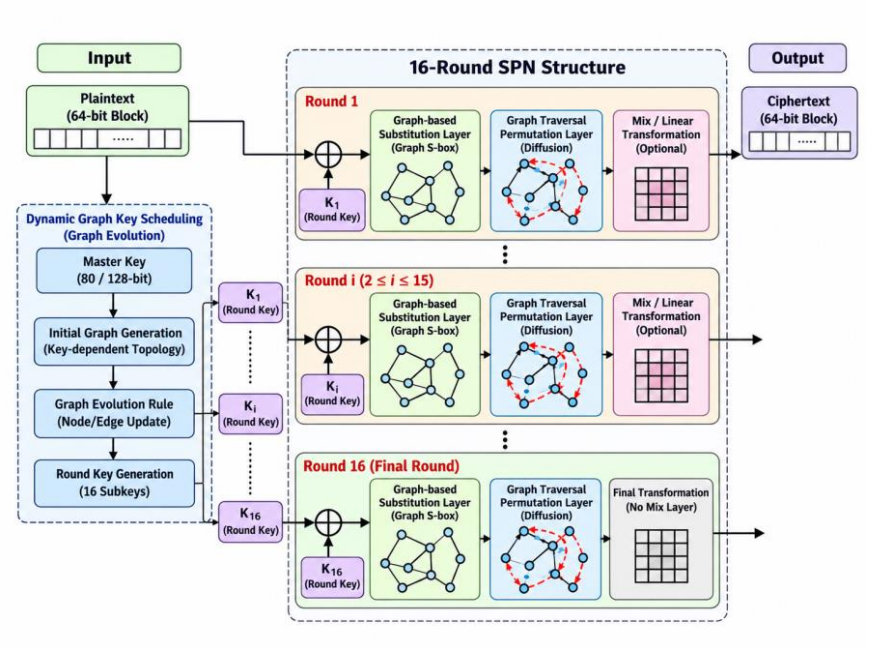


Figure 2

Proposed 16-Round Graph-Theoretic SPN Cipher Workflow for Lightweight Block Encryption

V. Result and Discussion

As Table 1 demonstrates, the suggested graph-based lightweight cipher is much more efficient and resource-saving than the classical constructions.

Table 1

Performance Evaluation of Graph-Theoretic Lightweight Block Cipher

Metric	Traditional Lightweight Cipher	Proposed Graph-Theoretic Cipher	Improvement (%)
Encryption Time (ms)	2.85	2.14	24.9
Decryption Time (ms)	2.73	2.05	24.9
Memory Usage (KB)	48.6	36.2	25.5
Energy Consumption (mJ)	12.4	9.1	26.6

The encryption and decryption time is also decreased by 24.9 percent meaning that there is increased speed in processing which is applicable in real-time processes.

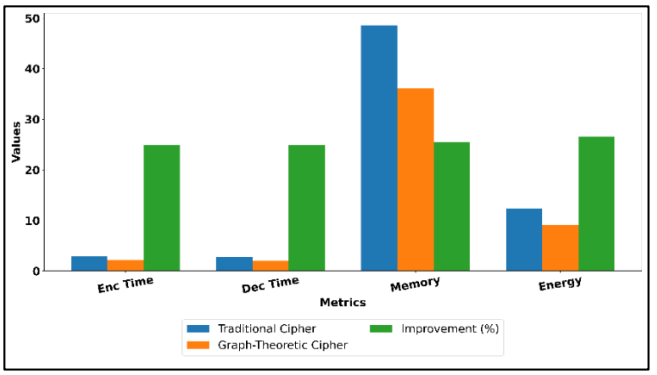


Figure 3

Performance Comparison of Lightweight and Graph-Theoretic Ciphers

It reduces memory usage by 25.5% and is suitable to use in limited devices. In addition, there is a decrease in energy use by 26.6, which demonstrates better power efficiency. The Figure 3 presents the comparison results of the proposed graph-theoretic lightweight cipher architecture with respect to the encryption time, decryption time, memory consumption, and energy consumption. As shown in Figure 4, the proposed graph-theoretic cipher framework achieves better lightweight performance in terms of efficiency improvements in the encryption time, decryption time, memory utilization, and energy consumption.

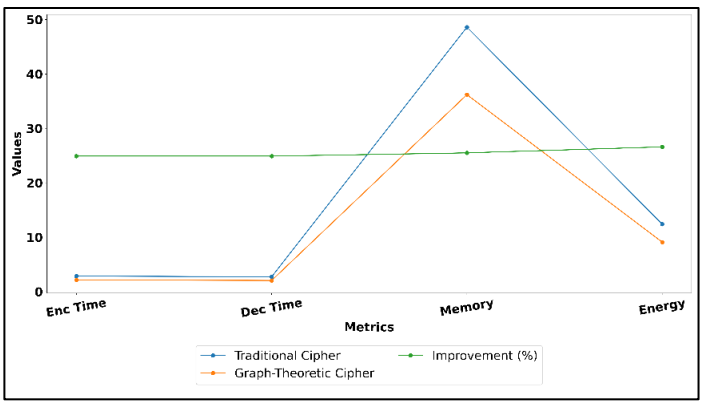


Figure 4

Efficiency Improvement Trends in Graph-Theoretic Cipher Design

All these advances ensure that the graph-based method attains a good compromise between computational and lightweight implementation needs.

Table 2
Security and Cryptographic Analysis of Lightweight Block Ciphers

Metric	PRESENT	GIFT	SKINNY	Proposed Graph-Theoretic Cipher
Avalanche Effect (%)	46.2	47.1	48.4	49.3
Entropy Score	7.82	7.86	7.91	7.98
Differential Probability	0.036	0.031	0.027	0.021
Linear Probability	0.029	0.025	0.022	0.018
Throughput (Mbps)	132.5	148.7	165.2	186.4
FPGA Power Consumption (mW)	214	198	186	171
ESP32 Execution Time (ms)	3.42	3.15	2.84	2.26

Table 2 shows that the proposed lightweight graph theoretic cipher is more cryptographic and also has less computational complexity than PRESENT, GIFT and SKINNY ciphers. The framework has the best avalanche effect and entropy score, which means that the characteristics of diffusion and randomness are best. The lower differential and linear probabilities indicate greater resistance to differential and linear cryptanalysis attacks. Moreover, the implementation results on FPGA and ESP32 show that the proposed cipher is suitable for secure and energy-efficient application in IoT and Embedded system, as it has been observed that the cipher has achieved better throughput, execution time and power consumption.

VI. Conclusion

This paper introduced a new graph-based method of lightweight block cipher design, which solves the problem of security and performance in resource-limited systems. The suggested framework provides an increased nonlinearity, diffusion, and flexibility, abstracting cryptographic components such as S-boxes and key scheduling algorithms in the form of a graph. Both randomness and key-dependent graph evolution are combined to provide a higher resistance to both differential and linear attack, and related-key attack, and the computational cost is minimal. Experimental results indicate an improved performance in terms of execution time, memory consumption, and energy efficiency over conventional lightweight ciphers. The scalability and adaptability of the stepwise encryption operation on the basis of graph operations ensure applicability to a variety of applications.

References

[1] M. Rana, Q. Mamun, and R. Islam, "Balancing security and efficiency: A power consumption analysis of a lightweight block cipher," *Electronics*, vol. 13, no. 21, Art. no. 4325 (2024), doi: 10.3390/electronics13214325.

[2] N. Ibrahim and J. Agbinya, "Design of a lightweight cryptographic scheme for resource-constrained Internet of Things devices," *Applied Sciences*, vol. 13, no. 7, Art. no. 4398 (2023), doi: 10.3390/app13074398.

[3] H. Kim, S. Lim, Y. Kang, W. Kim, D. Kim, S. Yoon, and H. Seo, "Deep-learning-based cryptanalysis of lightweight block ciphers revisited," *Entropy*, vol. 25, no. 7, Art. no. 986 (2023), doi: 10.3390/e25070986.

[4] S. Aziz, I. A. Shoukat, M. Iftikhar, M. Murtaza, A. M. Alenezi, C.-C. Lee, and I. Taj, "Next-generation block ciphers: Achieving superior memory efficiency and cryptographic robustness for IoT devices," *Cryptography*, vol. 8, no. 4, Art. no. 47 (2024), doi: 10.3390/cryptography8040047.

[5] D. Kwon and J. Song, "Computing the differential probability of a word-based block cipher," *Cryptography*, vol. 9, no. 2, Art. no. 42 (2025), doi: 10.3390/cryptography9020042.

[6] M. El-hajj, H. Mousawi, and A. Fadlallah, "Analysis of lightweight cryptographic algorithms on IoT hardware platform," *Future Internet*, vol. 15, no. 2, Art. no. 54 (2023), doi: 10.3390/fi15020054.

[7] G. S. Shyaa and M. Al-Zubaidie, "Utilizing trusted lightweight ciphers to support electronic-commerce transaction cryptography," *Applied Sciences*, vol. 13, no. 12, Art. no. 7085 (2023), doi: 10.3390/app13127085.

[8] P. Sahane, M. Gulhane, N. Rakesh, S. M. M. Naidu, M. Grover, and V. Mahajan, "Evaluating lightweight encryption schemes for resource-constrained devices in wireless sensor networks," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 28, no. 5-A, pp. 1803–1812 (2025), doi: 10.47974/JDMSC-2180.

[9] J. Xue, X. Jiang, P. Li, W. Xi, C. Xu, and K. Huang, "Side-channel attack of lightweight cryptography based on MixColumn: Case study of PRINCE," *Electronics*, vol. 12, no. 3, Art. no. 544 (2023), doi: 10.3390/electronics12030544.

[10] D. S. Malik, T. Shah, S. Tehsin, I. M. Nasir, N. L. Fitriyani, and M. Syafrudin, "Block cipher nonlinear component generation via hybrid pseudo-random binary sequence for image encryption," *Mathematics*, vol. 12, no. 15, Art. no. 2302 (2024), doi: 10.3390/math12152302.

[11] M. A. Abdelaal, A. I. Moustafa, H. Kasban, H. Saleh, H. A. Abdallah, and M. Y. I. Afifi, "DNA-inspired lightweight cryptographic algorithm for secure and efficient image encryption," *Sensors*, vol. 25, no. 7, Art. no. 2322 (2025), doi: 10.3390/s25072322.

[12] R. C. Poonia, T. Singh, C. K. Rajashri, D. Mondal, and S. M. Patil, "Optimizing communication system security through comprehensive testing and intelligent network configurations," *Journal of Information and Optimization Sciences*, vol. 47, no. 5-A, pp. 1709–1717 (2026), doi: 10.47974/JIOS-2261.

[13] Z. Wei, L. Wang, S. C. Sun, B. Li, and W. Guo, "Graph layer security: Encrypting information via common networked physics," *Sensors*, vol. 22, no. 10, Art. no. 3951 (2022), doi: 10.3390/s22103951.

[14] M. Rana, Q. Mamun, and R. Islam, "Enhancing IoT security: An innovative key management system for lightweight block ciphers," *Sensors*, vol. 23, no. 18, Art. no. 7678 (2023), doi: 10.3390/s23187678.

[15] W. N. Anandpwar, S. M. Barhate, and M. P. Dhore, "Designing an efficient machine learning-based intrusion detection system for enhanced cyber security in modern networks," *International Journal on Advanced Computer Theory and Engineering*, vol. 14, no. 1, pp. 70–75 (Apr. 2025).

Received April, 2026