Lightweight Block Cipher for Security in Resource-Constrained Network

Original Scientific Paper

Aruna Gupta*
CSE, Sathyabama Institute of Science and Technology,
Chennai, India
agupta7.2018@gmail.com

T. Sasikala
CSE, Sathyabama Institute of Science and Technology,
Chennai, India
dean.computing@sathyabama.ac.in

*Corresponding author

Abstract *– As the proliferation of resource-constrained devices continues in various application domains, the need for energy-efficient cryptographic algorithms becomes paramount for ensuring their security. Lightweight block ciphers play a crucial role in securing communication and data integrity in resource-poor environments. This paper presents the design, simulation, and evaluation of a novel symmetric Energy Efficient Lightweight Block Cipher (EE-LBC), tailored for such environments, which employs a balanced combination of substitution-permutation network (SPN) structure with larger diffusion and substitution box activation properties to achieve high security with minimal energy consumption and implementation cost. Through rigorous cryptanalysis and performance evaluations, EE-LBC demonstrates superior throughput and efficiency compared to prevailing lightweight block ciphers, making it an ideal choice for securing resource-constrained network.*

Keywords*: block cipher, IoT, lightweight cryptography, MANET, resource-constrained network*

Received: March 15, 2025; Received in revised form: April 14, 2025; Accepted: April 19, 2025

1. INTRODUCTION

With the rapid expansion of the Internet of Things (IoT) and the increasing integration of connected devices into various domains, ensuring data security in resource-constrained environments has become a paramount concern. Wireless networks having limited resources for storage, processing, communication and power are all categorized as Resource-Constrained Network (RCN) such as Mobile Ad-hoc NETwork (MANET), Vehicular Ad-hoc NETwork (VANET), Flying Ad-hoc NETwork (FANET) and Internet of Things (IoT). Nodes in such network must always retain minimum energy level so as to maintain the network connectivity [1]. On the other hand, security is the biggest challenge in these networks due to wireless communication media and lack of in-built security. Attacks done by the malicious node in the network can destructively affect the integrity, confidentiality, and secrecy of nodes in the network [2].

To guard the resource-constrained networks from active attacks, various proposals are available which are either proactive or reactive in nature. After the intruder disturbs the network, sense the attack and then try to recover from it. This is called a reactive method of protection. On the other hand, in proactive method, necessary care is taken to confirm that the intruder will not be able to damage the system or authentic user. Intrusion Detection Systems (IDS) are built on the reactive approach of defense whereas cryptography is preferred in the proactive method. One of the disadvantages of IDS is that it cannot detect the source of the attack and it just locks the whole network. This parallelizes it completely [3]. Secondly, IDS continuously monitor node behavior and network traffic; so, it keeps engaging the resources [4].

Proactive technique of cryptography can be split into two categories: cryptographic algorithms designed for resource-rich networks are not suitable for RCN due to their high resource requirements and other one is lightweight cryptographic primitives that uses limited resources without compromising the security level achieved [5, 6]. In this paper, we propose a novel en-

ergy-efficient lightweight block cipher, EE-LBC, specifically designed to address the security and energy constraints for application in the resource-poor networks.

The block cipher proposed in this paper is energy efficient as well secure and possesses following properties:

- Symmetric key cryptography-based block cipher EE-LBC uses Substitution-Permutation Network (SPN) structure as its base which comprises of substitution and permutation layer.
- High diffusion and S-box activation properties of EE-LBC makes it strongly resistance against differential and linear cryptanalysis.
- After taking the 80-bit key through key generation algorithm, the cipher assured to be resistant against key schedule attacks due to the shuffling of bits in the key by XORing round counter with the middle portion of the round key.
- It makes use of a single 4-bit non-linear S-box that operates on 16-bit data at a time. This leads to the simplicity in design while minimizing the computational complexity and implementation cost but improving throughput.
- Reduced number of rounds and smaller key size of this cipher results in desired performance in terms of energy efficiency.
- It is suitable for the constrained devices and networks for variety of applications.

Further portion of the paper is systematized by elaborating on distinguishable facts about various lightweight cryptographic protocols in Section II. Overview with the architecture and details of proposed block cipher are presented in Section III. Cryptanalysis of the proposed protocol is covered in Section IV whereas results of the comparison of EE-LBC with other block cipher and the discussion through analysis of the same is conversed in Section V. Paper is summarized in the Section VI followed by the references used.

2. RELATED WORK

For the past decade, various researchers have been engaged in discussing about the resource-constrained wireless networks and the security facets in its context. Precisely, the network layer attacks on routing protocols are a subtle issue in this and any other network [7, 8]. Traditional cryptographic algorithms are not suitable in this case due to the high resource requirements. Lightweight Cryptography is the method of encryption that leads to small-sized and computationally lower complexity as well as low power consumption. Thus, it extends the battery life of resource-limited devices while maintaining strong security levels. [9]. Lightweight cryptography consists of cryptographic protocols customized for implementation in constrained setup. Its standardization process is still in progress [10].

A contest held by NIST (Mar 2019-2021), to identify secure and efficient cryptographic algorithms suitable for constrained environments such as IoT devices and RFID systems, was graced by 57 innovative submissions out of which 56 could pass through the first round and 32 could reach to the final round of the contest. On the critical evaluation of those protocols, 10 were declared as finalists [11].

Prior research in lightweight cryptography has led to the development of various algorithms optimized for resource-constrained devices. Stream ciphers, block ciphers, and hash functions have been extensively studied and tailored for lightweight applications. One of the symmetric key cryptographic techniques for providing the confidentiality and integrity to the sensitive information is block cipher. Block cipher uses combination of confusion and diffusion properties of cryptography that makes the reverse of encryption process to extract the original text harder in block cipher [12]. Existing lightweight block ciphers, such as PRESENT [13], SIMON, SPECK [14] and LEA [15], have demonstrated promising results in terms of area efficiency and computational performance [16, 17]. However, these ciphers may still consume significant energy when implemented on power-constrained devices. In this section, prominent lightweight cryptographic block ciphers in this framework are studied and analyzed for security.

Two most popular examples of protocols standardized by NIST are AES [18] and DES [19]. Former is SPN based algorithm whereas later follows Feistel Network (FN) structure. AES has the implementation requirement of around 2400 GEs where DES needs around 2310 GEs. Such larger area pre-requisite makes both protocols unsuitable for RCN.

TWINE presented in [20] takes 64-bit input and has two variants with 80-bit and 128-bit key. It can operate with lesser memory also but has requirement of around 2000 GEs which is still higher for constrained environment. Ultra-light block cipher RECTANGLE in [21] works with least rounds i.e., 25 by applying little alterations to the SPN structure which makes it useful for large variety of applications in constrained environment. Ill-advisedly, it is suspectable to related-key and side-channel attacks.

Symmetric key block cipher E3LCM proposed in [22] uses Multi-sequence Linear Feedback Shift Register (MLFSR) in substitution layer for reducing design area. Though it has smallest key of 64-bit, since the output of MLFSR is deterministic, it is not secure.

PRINT [23] is another cipher that makes use of an 80-bit key to perform 48 iterations with least GE requirement. It performs 3-bit operations which is infeasible due to odd number of bits. On the other hand, PRINCE in [24] is one of most efficient lightweight algorithms that uses 128-bit key for 12 rounds. It has low energy consumption and smaller hardware requirement. But it is not used widely due to its fixed larger key size.

Major focus of TEA in [25] is to achieve higher speed while minimizing the memory requirements. It is designed to work for commodity hardware with the application of 128-bit key used across 32 rounds. But its key scheduling part is too simple to get forged by brute force attack. SKINNY [26] has three key variants 64/128/192-bit key to perform 32 to 40 rounds on the data blocks of varying sizes. Limitation of this cipher is that it is prone to birthday attack.

GIFT [27] was finalist in NIST contest since it offers lighter S-box and occupies lesser physical space. It uses 128-bit key for 64-bit data block with 40 rounds. But it's not safe against differential cryptanalysis. SLIM [28] is used for Internet of Health Things and is based on Feistel structure. It works on 32-bit block with 80-bit key in 32 rounds. Its security is suitable only for RFID-based systems. It is immune against linear and differential cryptanalysis.

The implementation of the KATAN32 algorithm [29] on the ESP32 microcontroller demonstrates low computational power requirements, making it suitable for resource-constrained environments. While KATAN32 is lightweight, it may not offer the same level of security robustness as more modern algorithms. The implementation is tailored to the ESP32 microcontroller, which may limit its generalizability to other hardware platforms without additional modifications.

Lastly, ASCON [30], selected as the NIST lightweight cryptography standard, requires significantly fewer resources compared to AES-128, making it well-suited for edge devices. It has been found resistant against side-channel attacks, a critical consideration for IoT devices. Its limitation is that performance metrics such as resource utilization, operating frequency, and power consumption can vary across different FPGA platforms, necessitating platform-specific optimizations.

As seen from the above study, SLIM, KATAN32 and ASCON have platform-specific requirements. AES, DES, and Twine ciphers exhibit significant area overhead, making them resource-intensive for hardware implementations. E3LCM, PRINT, and TEA ciphers exhibit inherent structural vulnerabilities, impacting their cryptographic robustness. A block cipher is expected to provide balance among implementation cost, performance in terms of encryption time and security.

3. EE-LBC ALGORITHM

The algorithm EE-LBC is predicated on Substitution-Permutation Network (SPN) structure that shudders the data through a combination of substitution layer and permutation layer and puts it together for the further round. It spins the data through the 31 rounds of permutations with the assistance of separate roundkey for every round. The encryption method of proposed cryptographic protocol accepts two inputs, one data block of size 64-bit and a key stream that is 80-bit long and generates ciphertext block as an output. During each of the 31 rounds, XOR operation is performed between the data block and the corresponding roundkey. In each round, the non-linear Substitution Layer that consists of 4-bit S-box is applied 16 times (64-bit data block/4-bit S-box=16) parallelly. Decryption process of this cipher is the reverse of steps applied during encryption. The diagram showing functioning of proposed algorithm EE-LBC is shown in Fig. 1.

The resistance of symmetric-key based block ciphers broadly depends on the cryptographic potency of the Substitution Layer. To scale back the design area, the Substitution Box structure is slightly altered and constructed using 4-bit single S-box rather than eight S-boxes. The rifeness of this structure is that it occupies less space with finest speed and energy consumption. The research work in this paper focuses solely on a software-based implementation and analysis, which aims to demonstrate the fundamental design, correctness, and performance of the proposed cipher on constrained devices. Implementation of EE-LBC is focused on embedding a lightweight cryptographic protocol in routing protocol of MANET-enabled IoT. Detailed stages in the form of bit operations performed during encryption and decryption processes of the proposed block cipher are delineated below.

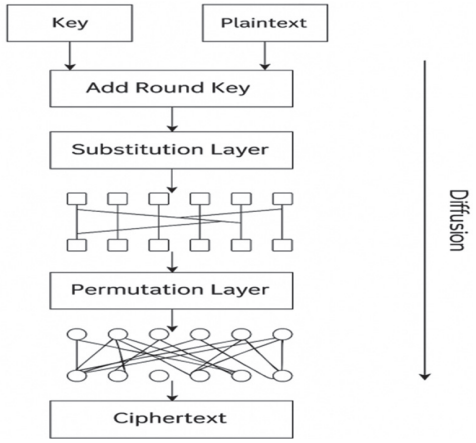

Fig. 1. Block Diagram for EE-LBC

A. Encryption

Key Scheduler

EE-LBC incorporates a lightweight key scheduling algorithm that minimizes computational overhead and energy consumption during key expansion. The key scheduling algorithm efficiently generates round keys from the master key, ensuring robust key mixing without sacrificing performance.

One key K_i is generated for each round i where 1<= i <= 31. At first, the original 80-bit key (K_{79}, K_{78}……K_0) is split into two keys, namely Key16 which is formed with lower 16 bits of the key and Key64 which is made up of upper 64 bits of the key.

$$\text{Key16} = (K_{15}, K_{14}\ldots\ldots K_0)$$
$$\text{Key64} = (K_{79}, K_{78}\ldots\ldots K_{16})$$

For round 1 the key K_1 is simply the Key64. Remaining round keys are generated using following steps:

- Shift the key 61 bits to the left: $K_i = K_i << 61$
- Key64 goes through the substitution layer
- The bits at the mid of the two keys are XORed with the counter of the current round

$$(K_{19}, K_{18}, K_{17}, K_{16}, K_{15}) = (K_{19}, K_{18}, K_{17}, K_{16}, K_{15}) \oplus i$$

For each round, the upper 64-bit part of the generated key is used as the round key.

After generating subkeys for all rounds, the encryption algorithm iterates through following three phases 31 times:

Add Round Key

The round function combines the operations of the substitution and permutation layers with the round key to produce the output ciphertext block. It consists of multiple iterations of the substitution and permutation operations followed by key mixing.

The 64-bit data block is XORed with the 64-bit round key Ki generated by the key scheduler as follows:

$$(B_{63}, B_{62}\ldots\ldots B_0) = (B_{63}, B_{62}\ldots\ldots B_0) \oplus (K_{63}, K_{62}\ldots\ldots K_0)$$

Substitution Layer

The simple design of S-box in EE-LBC focuses on achieving a balance between cryptographic strength and reduced computational complexity. This also maintains sufficient non-linearity and resistance against differential and linear cryptanalysis. In EE-LBC, single 4-bit S-box is used. The above data block after addition of round key is passed through the substitution layer. The S-box is defined with the hexadecimal values as follows:

S[] = {0xC, 0x5, 0x6, 0xB, 0x7, 0x0, 0xA, 0xD, 0x1, 0xE, 0xF, 0x8, 0x4, 0x9, 0x3, 0x2}

Each word (4-bit) in the data block is replaced by the corresponding value in the S-box at the same position.

Permutation Layer

The permutation layer shuffles the output of the substitution layer to achieve diffusion. It ensures that each bit of the input affects multiple bits of the output, thereby spreading the influence of each input bit across the entire block. This layer is implemented using efficient permutation techniques to minimize energy consumption.

For performing permutation at each round, following table (Table 1) is referred:

Table 1. Permutation Table

Bit Position 1															
0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63
NewBit Position														Bit Position 64	

Reading the table row-wise, each cell number represents, input bit position as marked on the table. So, ith bit is moved to the position indicated by the value inside the cell.

During each round of the encryption process, the permutation layer performs a series of bit-level operations designed to enhance diffusion by disrupting bit positions systematically. The steps are as follows:

1. Bit Position Identification: Determine the position of the bit to be processed by calculating its distance from the least significant bit (LSB). This helps in isolating its exact contribution to the data block.

2. Bit Alignment: Right shift the data block to align the target bit with the LSB position. This normalization simplifies subsequent operations.

3. Bit Isolation: Apply a bitwise AND operation with 1 to mask and extract the target bit. This ensures only the bit of interest is manipulated.

4. New Position Calculation: Compute the target location for the bit based on a permutation rule or key-driven logic. This step ensures that the permutation is non-linear and key-dependent.

5. Bit Placement: Shift the isolated bit to its new position and use a bitwise OR operation to merge it into the permuted data block. This ensures that each bit contributes uniquely to the final encrypted output.

B. Decryption

Key scheduler module, adding round key and substitution layer for the decryption process is same as that of encryption process. Last phase is replaced by inverse permutation layer. During each round of the decryption process, the inverse permutation phase systematically reconstructs the original bit positions to reverse the scrambling introduced during encryption. The following steps are executed:

1. Target Bit Identification: Determine the original position of the bit that should occupy the i-th position in the final output. This step ensures accurate reversal of the permutation logic.

2. Result Alignment: Left shift the intermediate result by 1 bit to create space for inserting the next bit in its correct sequence.

3. Bit Alignment in Cipher Block: Shift the data

block such that the required bit is brought to the least significant bit (LSB) position, simplifying its extraction.

4. Bit Extraction: Apply a bitwise AND operation with 1 to mask and isolate the required bit from the shifted data block.

5. Bit Integration: Insert the extracted bit into the result using a bitwise OR operation. This reconstructs the inverse permutation step-by-step, ultimately restoring the original data structure.

4. CRYPTANALYSIS OF EE-LBC

The attacks that alter the data or affects the memory typically catches hold of parameters of block cipher for performing the attack and they do not exploit the inner structure of the block cipher. In further part of this section, crucial security measure is discussed in context to the proposed technique. Cryptanalysis is employed to assess how resistant the algorithm is to various types of attacks. Two most commanding tools for cryptanalyst for finding linear or differential path through various rounds of the block cipher are linear and differential cryptanalysis [31]. But if the search space of the cipher is sufficiently large then finding the optimal path becomes challenging task for the cryptanalyst.

Differential Cryptanalysis:

During this technique, attacker selects the plaintext as input to the algorithm and then access the corresponding ciphertext. To perform this cryptanalysis, a pair of plaintexts is taken which is related with each other by a constant difference. It exploits how difference in plaintext pairs propagate through the cipher's structure [32].

The difference Δc between the generated pair of ciphertexts for the given pair of plaintexts can be calculated as

$$\Delta c = S\,(P \oplus \Delta p) \oplus S\,(P).$$

where S is the substitution function and P is the permutation function, Δp indicates the difference between the given pair of plaintexts.

Consider following pair of inputs with just one-bit difference:

P1: 4172756e61204b47

P2: 4172756e61204b48

It generates following pair of ciphertexts with countable difference:

C1: fff2f964d2012604

C2: 5a3c5405043f0d45

The 4-bit S-box used in EE-LBC has a maximum differential probability (MDP) of 2^{-2}. Since the cipher has 16 S-boxes per round and 31 rounds, the best differential characteristic over many rounds involves selecting active S-boxes carefully.

- Over 5 rounds: at least 10 active S-boxes.

- Over 25 rounds: ≥ 62 active S-boxes.

If 62 S-boxes are active and each contributes max 2^{-2}, total probability ≤ $(2^{-2})^{62} = 2^{-124}$. So the cipher resists differential attacks beyond about 25 rounds with time complexity of around 2^{63} encryptions.

Linear Cryptanalysis:

Linear estimate showing relation between plaintext and ciphertext is computed in linear cryptanalysis to analyze the block cipher [33]. We can calculate the linear trails through various rounds of the algorithm. Linear trail is calculated by producing the linear estimates of the S-box.

As per theorem in [34] for linear estimates, like in the differential case, let's say 62 active S-boxes:

Total linear trail bias: $(2^{-1})^{62} = 2^{-62}$.

Required plaintexts ≈ 1 / $(\text{bias})^2 = 2^{124}$

This is again infeasible with cipher's 64-bit block size. Best known linear attack reaches up to 26 rounds with time complexity ~2^{65} and data complexity ~2^{63}.

Also, diffusion property of the bit permutation used in EE-LBC as explained in section III is strong enough to defend against linear cryptanalysis.

Key Schedule Attack:

By shuffling the bits of the key with the help of non-linear operations, the block cipher EE-LBC can resist key schedule attacks like round key attack where the intruder tries to identify the relationship between various sets of subkeys. This is done in the third step of key scheduler module by using round counter that is XORed with the middle portion of the round key. While generating each of the round key, use of non-linear function is made as described in the section III.

Although the current implementation of EE-LBC is software-based and not evaluated for side-channel resistance, potential countermeasures such as constant-time operations and masking techniques can be considered in future hardware implementations to enhance resilience against side-channel attacks.

To summarize the methodology of EE-LBC for attack resistance, while generating round keys, shuffling bits in the key using non-linear function by XORing round counter with the middle portion of the round key, makes it possible to protect the cipher from key schedule attacks. Due to its proper S-box activation properties, it has strong resistance against differential cryptanalysis. Efficient bit permutation leads to high diffusion which directly strengthens its security against linear cryptanalysis.

5. RESULTS AND DISCUSSION

The core goal of the proposed protocol of designing simplistic solution for security of resource-constrained network is consummated by optimizing the implementation for the performance. The key size of EE-LBC that is 80-bit provides more than acceptable security for the

applications requiring basic security, typically the one that uses tag-based deployments like applications in health-care sector, smart agriculture based on IoT etc.

Experimental Set-up:

The proposed protocol is implemented and executed using the simulator Omnet++ 5.7 [35, 36] to test its functionality. This simulator offers open-source framework based on C++ library. It supports for the communication of mobile and wireless network nodes [37]. The base network intended for the experiment is MANET which is wireless ad-hoc network. Simulation environment set-up indicating parameter values for the same are as follows:

Number of nodes	50
Number of connection links	varying
Broadcast delay	0.01 ms
Datarate	1 kbps
Project reference	queueinglib
Simulation runtime GUI	Qtenv
Simulation run mode	Fast
Data block size	64-bit
Message frequency	0s
Routing Protocol	AODV

As detailed in section II, several block ciphers have been examined, among which few with smaller key size and smaller data block processing per operation, have been simulated and further analyzed in this research for comparative purposes. These block ciphers along with EE-LBC as detailed in Table 2, were simulated using Omnet++. Comparative results of these ciphers are presented below. The comparison presented in Table 2 illustrates the security parameters of EE-LBC in relation to other block ciphers, including number of rounds, key size, throughput and immunity against cryptographic attacks etc.

Table 2. Security Analysis of Block Ciphers

Block Cipher	Key Length	Data Block Size	Rounds	No. of S-boxes	Throughput (Kbps)	Cryptanalysis
GIFT	128 bit	64 bit	28	08	20.40	Not safe against related-key attack
Twine	80 bit	64 bit	36	08	12.48	Not immune against related-key differential attack
PRINT	80 bit	48 bit	48	08	2.2	Resistant only to related key attack
KATAN32	80 bit	32 bit	254	-	3.5	Not safe against linear, differential and related-key attacks
ASCON	128 bit	64 bit	30	05	10	Resilient to differential and linear attacks
EE-LBC	80 bit	64 bit	31	01	23.7	Resistant against key scheduling attack, linear and differential cryptanalysis

Comparative Analysis:

Data Block Size and Key Size:

The 64-bit data block size is typical for lightweight ciphers, balancing between security and efficiency. The 80-bit key size provides a reasonable level of security for most practical applications but may be susceptible to brute-force attacks in the long term. GIFT and ASCON have a longer key compared to other ciphers that increases complexity during key scheduling. Relation of key-length with number of rounds for above block ciphers is presented in Fig. 2.

Throughput:

Throughput measures how quickly data can be processed. Higher throughput is generally favorable, especially in scenarios where real-time processing is critical. As compared to others, the cipher EE-LBC has higher throughput 23.7 Kbps due to its lesser block size, S-box and number of rounds, whereas PRINT has lowest of 2.2 Kbps. Comparison of throughput for all above ciphers is shown in Fig. 3 below.

Number of S-boxes:

The number and design of S-boxes contribute significantly to the security and cryptographic strength of the cipher. More S-boxes can enhance security against certain types of attacks but might increase computational complexity. Where GIFT, Twine and PRINT use 8 S-boxes, EE-LBC lowers the complexity by applying single 4-bit S-box 16 times.

Cryptanalysis Resistance:

This parameter assesses how resilient the cipher is against known attacks, such as differential and linear cryptanalysis, which are common in the evaluation of block ciphers. GIFT is strong against linear cryptanalysis due to its robust S-boxes and balanced diffusion properties. Twine and ASCON are resilient against differential and linear cryptanalysis through their strong S-boxes, round structure, and key schedule design. PRINT and KATAN32 are not immune against these cryptanalyses. Whereas EE-LBC is resistant to key scheduling attack, differential and linear cryptanalysis through its non-linear layers, permutation layer, complex key schedule and careful round function design as conferred in Section IV.

Encryption Time:

The simulation encompassed various message ranges, recording the encryption process time for the above ciphers depicted in Fig. 4. Encryption time is inversely proportional to the throughput. For comparison, we consider the simulation run for 100 messages, each of

size 1KB (8000 bits). By referring the throughput values from Table 2, encryption time for the above ciphers is computed using following formula:

Encryption Time = (Block Size (bits) x Number of Messages) / Throughput (bps)

Graph portraying the same in Fig. 4 clearly shows that EE-LBC requires the least encryption time of 33.76s and PRINT is ahead of the other ciphers due to the least throughput.

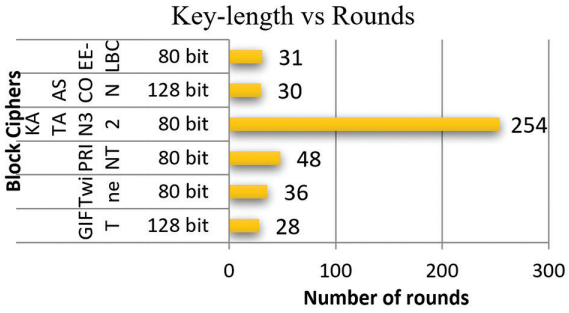

Fig. 2. Number of Rounds for ciphers

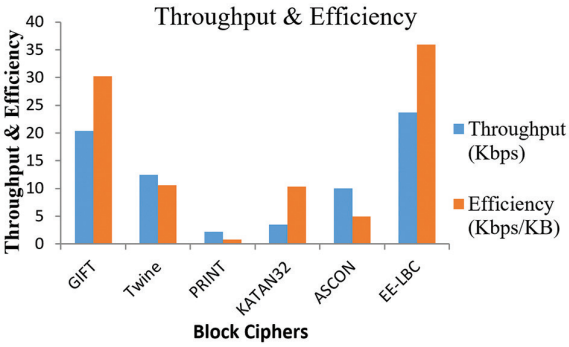

Fig. 3. Throughput and Efficiency of ciphers

Efficiency:

Another essential metric efficiency provides performance while minimizing resource requirements. It is dominated by the lengthier algorithms. It can be determined as follows:

Efficiency = Throughput (Kbps) / Code_Size (KB)

Graph in Fig. 3 for efficiency of the block ciphers is led by EE-LBC with the peak efficiency 35.91 Kbps/KB due to its smaller code size and trailed by PRINT with its extensive code.

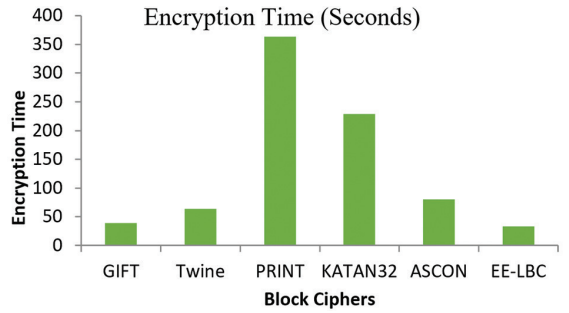

Fig. 4. Time for Encryption Process

6. CONCLUSION

The study focuses on securing resource-constrained wireless networks, which are vulnerable to attacks like man-in-the-middle that threaten control message integrity. The key challenge is developing energy-efficient security solutions, as conventional cryptographic algorithms are too resource-intensive. Lightweight cryptography offers a viable alternative with lower power and computational demands.

This paper introduces design and simulation of EE-LBC, a symmetric lightweight cryptographic block cipher structured on SPN), operating on a 64-bit data block with an 80-bit key, swirling through 31 rounds. The algorithm prioritizes simplicity in design by dropping S-box count to one and reduction in implementation cost while ensuring a satisfactory level of security making it an ideal choice for securing IoT devices and other energy-constrained systems. The algorithm exhibits resilience against key schedule attack, algebraic attack as well as linear and differential cryptanalysis. However, there remains a marginal vulnerability to biclique attacks, contingent upon the attacker conducting exhaustive computations involving approximately 2^{80} encryption attempts to determine the correct key.

Performance evaluation clarifies significant rise in throughput i.e., 23.7 Kbps whereas there is reduction of encryption time, for 100 messages each of size 1KB, of 13.92% compared to GIFT with least encryption time among others. Future work may involve further optimizations and extensions to EE-LBC, as well as exploration of its applicability to emerging IoT scenarios and use cases.

7. REFERENCES:

[1] Q. V. Khanh, L. A. Ngoc, "An energy-efficient routing protocol for MANET in Internet of Things environment", International Journal of Online and Biomedical Engineering, Vol. 17, No. 7, 2021, pp. 35-45.

[2] M. A. Khan, I. M. Qureshi, F. Khanzada, "A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (FANET)", Drones, Vol. 3, No. 1, 2019, p. 16.

[3] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, R. Jain, "Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study", IEEE Access, Vol. 8, 2020, pp. 106576-106584.

[4] J. Jabez, B. Muthukumar, "Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach", Procedia Computer Science, Vol. 48, 2015, pp. 338-346.

[5] A. Biswas et al. “LRBC: a lightweight block cipher design for resource constrained IoT devices”, Journal of Ambient Intelligence and Humanized Computing, Vol. 14, 2023, pp. 5773-5787.

[6] Y. Zhong, J. Gu, “Lightweight block ciphers for resource-constrained environments: A comprehensive survey,” Future Generation Computer Systems, Vol. 157, 2024, pp. 288-302.

[7] B. V. dos Santos, A. Vergütz, R. T. Macedo, M. Nogueira, “A dynamic method to protect user privacy against traffic-based attacks on smart home”, Ad Hoc Networks, Vol. 149, No. C, 2023.

[8] S. Ganesh, R. Amutha, “Efficient and secure routing protocol for wireless sensor networks through SNR based dynamic clustering mechanisms”, Journal of Communications and Networks, Vol. 15, No. 4, 2013, pp. 422-429.

[9] F. Dridi, S. El Assad, W. El Hadj Youssef, M. Machhout, R. Lozi, “Design, implementation, and analysis of a block cipher based on a secure chaotic generator”, Applied Sciences, Vol. 12, No. 19, 2022, p. 9952.

[10] O. Toshihiko, “Lightweight cryptography applicable to various IoT devices”, NEC Technical Journal, Vol. 12, No. 1, 2017, pp. 67-71.

[11] NIST Contest, “Lightweight Cryptography: Finalists and Standardization”, https://csrc.nist.gov/projects/lightweight-cryptography (accessed: 2025)

[12] M. B. İlter, “Differential and Linear Cryptanalysis of Lightweight Block Ciphers with MILP Approach”, Graduate School of Applied Mathematics, Middle East Technical University, Ankara, Turkey, 2023, Ph.D. thesis.

[13] A. Bogdanov et al. “PRESENT: An Ultra-Lightweight Block Cipher”, Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, Austria, 10-13 September 2007.

[14] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, "The SIMON and SPECK Families of Lightweight Block Ciphers”, IACR Cryptology ePrint Archive, Vol. 2013, 2013, p. 404.

[15] D. Hong et al. "LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors", Proceedings of the 14th International Workshop on Information Security Applications, Jeju Island, Korea, 19-21 August 2013, pp. 3-27.

[16] V. A. Thakor, M. A. Razzaque, M. R. A. Khandaker, "Lightweight Cryptography for IoT: A State-of-the-Art", IEEE Internet of Things Journal, Vol. 7, No. 10, 2020, pp. 9370-9383.

[17] V. A. Thakor, M. A. Razzaque and M. R. A. Khandaker, “Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities”, IEEE Access, Vol. 9, 2021, pp. 28177-28193.

[18] N. Pub, “Advanced encryption standard (AES)”, Federal Information Processing Standards, Vol. 197, No. 441, 2001, p. 0311.

[19] R. Anusha, V. V. D. Shastrimath, “LCBC-XTEA: High throughput lightweight cryptographic block cipher model for low-cost RFID systems”, Proceedings of the 8th Computer Science On-line Conference: Cybernetics and Automation Control Theory Methods in Intelligent Algorithms, Vol. 3, 2019.

[20] T. Suzaki, K. Minematsu, S. Morioka, E. Kobayashi, “Twine: A lightweight, versatile block cipher”, Proceedings of the 19th International Conference on Selected Areas in Cryptography, Windsor, Canada, 15-16 August 2012, pp. 1-5.

[21] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, I. Verbauwhede, “RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms”, Science China Information Sciences, Vol. 58, No. 12, 2015, pp. 1-15.

[22] Periasamy et al. “An Enhanced Energy Efficient Lightweight Cryptography Method for various IoT devices”, ICT Express, Vol. 7, 2021.

[23] L. Knudsen, G. Leander, A. Poschmann, M. Robshaw, “Printcipher: A block cipher for IC-printing”, Proceedings of the 12th International Workshop, Santa Barbara, CA, USA, 17-20 August 2010, pp. 16-32.

[24] J. Borgho et al. “PRINCE—A low-latency block cipher for pervasive computing applications”, Proceedings of the 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, 2-6 December 2012, pp. 208-225.

[25] D. Williams, "The tiny encryption algorithm (TEA)", Network Security, Vol. 26, 2008, pp. 1-14.

[26] C. Beierle et al. "Sim.: The skinny family of block ciphers and its low-latency variant mantis", Proceedings of the 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, 14-18 August 2016, pp. 123-153.

[27] S. Banik et al. "Gift: A Small Present: Towards Reaching the Limit of Lightweight Encryption", Proceedings of the 19th International Conference on Cryptographic Hardware and Embedded Systems, Taipei, Taiwan, 25-28 September 2017.

[28] B. Aboushosha et al. "SLIM: A Lightweight Block Cipher for Internet of Health Things", IEEE Access, Vol. 8, 2021, pp. 203747-203757.

[29] A. Ukpebor, J. Addy, K. Ali, A. A. Humos, "Secure End-to-End Communications with Lightweight Cryptographic Algorithm", IEEE Internet of Things Journal, Vol. 10, No. 2, 2023, pp. 1023-1034.

[30] J. Kaur, A. C. Canto, M. M. Kermani, R. Azarderakhsh, "A Comprehensive Survey on the Implementations, Attacks, and Countermeasures of the Current NIST Lightweight Cryptography Standard", IEEE Access, Vol. 11, 2023, pp. 12345-12367.

[31] J. S. Teh, L. J. Tham, N. Jamil, W.-S. Yap, "New differential cryptanalysis results for the lightweight block cipher BORON", Journal of Information Security and Applications, Vol. 66, 103129, 2022.

[32] J. Lu, "A Methodology for Differential-Linear Cryptanalysis and Its Applications", Designs, Codes and Cryptography, Vol. 77, No. 1, pp. 11-48, Oct. 2015.

[33] S. Sallam, B. D. Beheshti, "A survey on lightweight cryptographic algorithms", Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea, 28-31 October 2018, pp. 1784-1789.

[34] A. Bogdanov, P. S. Vejre, "Linear Cryptanalysis of DES with Asymmetries", in ASIACRYPT 2017, Vol. 10624, Eds. Cham: Springer, 2017, pp. 187-216.

[35] OmNET++ 5.7, https://omnetpp.org/download/ (accessed: 2025)

[36] A. Varga and OpenSim Ltd. "OMNeT++ User Guide Version 6.0", https://omnetpp.org/doc/omnetpp5/UserGuide.pdf (accessed: 2025)

[37] S. Manzoor, M. Manzoor, H. Manzoor, D. E. Adan, M. A. Kayani, "Which Simulator to Choose for Next Generation Wireless Network Simulations? NS-3 or OMNeT++", Engineering Proceedings, Vol. 46, No. 1, 2023, p. 36.