Enabling low-latency IoT communication for resource-constrained devices with the led cipher and decipher protocol

Mahendra Shridhar Naik[1,4], Desai Karanam Sreekantha[2], Kanduri VSSSS Sairam[3]
[1]Department of Computer Science and Engineering, NMAM Institute of Technology, Nitte Affiliated to Visvesvaraya Technological University, Belagavi, India
[2]Department of Computer Science and Engineering, NMAM Institute of Technology, NITTE (Deemed to be University), Nitte, India
[3]Department of Electronics and Communication Engineering, NMAM Institute of Technology, NITTE (Deemed to be University), Nitte, India
[4]Department of Electronics and Communication Engineering, New Horizon College of Engineering, Bengaluru, India

Article Info

Article history:

Received Jan 25, 2024
Revised Feb 19, 2024
Accepted Feb 22, 2024

Keywords:

Decryption
Encryption
Light-encryption-device
Latency
Throughput

ABSTRACT

Block cipher algorithms are crucial for securing applications on resource-constrained devices. This paper introduces the modified light encryption device (MLED) cipher-decipher architecture, specifically designed to accommodate both 64-bit and 128-bit key sizes while maintaining a consistent 64-bit block and data size. MLED comprises 8-step and 12-step processes for MLED-64 and MLED-128 modules, respectively. Each stage involves a four-round operation followed by an add-round key operation. The add constant module (ACM) and mixed column modules (MCMs) within the round operation have been optimized for improved latency and throughput. Performance analysis reveals that MLED-64/128 requires less than 1% of the available slices and operates at 125 MHz on the Artix-7 FPGA. It achieves delays of 7.5 and 12.5 clock cycles for MLED-64 and MLED-128, respectively, translating to throughputs of 1366.5 Mbps and 819.89 Mbps. Additionally, MLED-64/128 exhibits hardware efficiencies of 2.373 and 0.986 Mbps/slice, respectively. Comparative evaluations with existing LED and other block ciphers (BCs) demonstrate that MLED-64/128 achieves significant improvements in latency, throughput, and efficiency, making it a compelling choice for securing resource-constrained IoT applications.

Corresponding Author:

Mahendra Shridhar Naik
Department of Computer Science and Engineering, NMAM Institute of Technology
Affiliated to Visvesvaraya Technological University, Nitte, Karnataka, India
Email: mahendrasnaik@gmail.com

1. INTRODUCTION

Numerous intelligent gadgets establish communication and interaction among themselves through sensors, utilizing the internet of things (IoT) infrastructure. This network enables users to access and share relevant information according to their own needs. IoT devices have been widely implemented across various domains, including healthcare, home automation, intelligent systems, surveillance, tracking systems, and other applications. IoT devices encounter numerous security difficulties, encompassing but not limited to authentication, privacy concerns, data protection, confidentiality, performance degradation resulting from assaults, and various others [1]. Block ciphers (BCs) are a viable solution for addressing many security concerns, particularly in resource-constrained devices such as IoT devices, wireless sensor networks (WSNs), body area networks (BANs), and radio-frequency identification (RFID) technologies. The boundary conditions can be classified into two main categories: conventional boundary conditions (CBCs) and lightweight boundary conditions (LBCs). The CBCs employ data sizes of either 64 or 128 bits, whereas the LBCs utilize data sizes of 32, 48, or 64 bits. The potential key sizes of these BCs may exhibit variation based on the specific requirements and the number of rounds involved [2], [3]. The boundary conditions are implemented using either a software-based or hardware-based technique in order to accommodate low-resource devices. The software method can be classified as either machine-independent or machine-dependent. On the other hand, the hardware approach is specifically developed with regard to the utilization of field-programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), or a full-custom methodology. The performance parameters associated with the hardware implementation of BCs encompass chip area, area cost, efficiency, latency, throughput, figure-of-merit (FOM), energy consumption, power consumption, and energy per bit [4]-[6].

Various architectures are employed in the construction of BCs, including the substitution-permutation network (SPN), a network based on the Feistel cipher, a technique utilizing a non-linear feedback shift register (NLFSR), and the Lai-Massey approach [7], [8]. The light encryption device (LED) BC is a good way to look into a number of security holes, such as side-channel attacks (both classical and quantum), key recovery attempts, and differential cryptanalysis [9], [10]. This article shows how to make a modified light encryption device (MLED) cipher-decipher architecture that works quickly, has low latency, and can handle a lot of data. The MLED module provides support for key sizes of both 64-bit and 128-bit, which can be utilized for data and block sizes of 64-bit. The significance of the planned MLED-64/128 is emphasized in the following manner: In order to optimize performance, the add-round constant values are utilized directly in round and inverse-round operations without the need for any left-shifting action. This approach helps reduce the number of clock cycles required. The four-round operations, along with the XOR operation with the key, are performed concurrently and within a single clock cycle, enhancing the overall latency of the system. The current mixed columns matrix is switched out for a more advanced encryption method called the advanced encryption standard (AES)-based mixed columns matrix for multiplication. This improves throughput and latency.

The manuscript's structure is delineated as follows: Section 1 provides a comprehensive discussion of the background of the planned work, with a focus on recent studies. The detailed explanation of the proposed MLED-64/128 hardware design and its sub-modules may be found in section 2. In section 3, there is an analysis of the simulation and synthesis results as well as a performance comparison. Finally, the study closes the entire body of work by highlighting advancements in the realm of future possibilities in section 4.

2. LITERATURE REVIEW

This section thoroughly examines the contextual foundation of the desired inquiry, focusing on current scholarly investigations. Bogdanov *et al.* [11] created "PRESENT" as a compact BC to meet the increasing need for RFID tags and affordable devices. The encryption consists of 31 rounds and 64-bit blocks and utilizes a substitution box to enhance cryptographic performance and efficiency. It provides protection against differential and linear cryptanalysis and has resilience against attacks. The hardware implementation necessitated at least 1,000 gate-equivalents, rendering it well-suited for RFID applications. The design of the present offers extremely lightweight cryptographic protection on a restricted number of devices and encourages community cryptanalysis to enhance security.

The LED [12] BC was created to meet the needs of secure data transfer in resource-limited scenarios, such as those in RFID tags and sensors. The LED employs a structure known as a substitution-permutation network (SPN). The cryptographic technique operates on data blocks that are 64 bits in size. The system offers assistance for key lengths of 64 bits and 128 bits. The cipher's predictability is due to its fixed number of iterations (48 or 64, depending on the key length), making the encryption process straightforward and allowing for estimation of the time it will take. The goal of LED is to provide a reasonable level of protection against regularly encountered cryptographic threats. Its structure's architecture is designed to efficiently combat both differential and linear cryptanalysis techniques. LEDs are advantageous because of their great efficiency. The design is meant to be executed with restricted hardware resources. This goal is achieved by utilizing elements and design principles from current lightweight designs and adapting them to enhance their usefulness for LED technology. LED technology is ideal for devices that cannot support complex encryption methods because of constraints in power consumption, physical space, or economic limitations.

Shibutani *et al.* [13], a very lightweight BC created for environments with limited resources such as RFID tags and sensors. The cipher seeks to achieve a balance between security, hardware efficiency, and power consumption by utilizing Feistel-based structures and F-functions. The encryption algorithm accommodates key lengths of 80 and 128 bits, with a consistent block length of 64 bits. Piccolo's security research shows that it is resilient to differential and linear cryptanalysis attacks. The study examines hardware and software solutions, with a focus on minimal area and efficiency measures. Piccolo is evaluated against other modern lightweight BCs in terms of security, area, and power consumption, demonstrating comparative performance and efficiency. Piccolo is a reliable choice for lightweight cryptographic primitives in embedded systems, providing strong security and versatility for many applications.

Yang *et al.* [14] lightweight BCs in 2015. The objective was to improve the security features of devices with limited resources, like IoT devices and low-power electronics. The authors aimed to create a distinctive series of BCs that optimized the balance between security, performance, and resource needs. The Simeck cipher, created by the National Security Agency (NSA), combines the positive characteristics of Simon and Speck, two earlier lightweight BCs. The cipher's circular design and flexibility to adjust to different security levels and application needs contribute to its lightweight nature. This work examines the security features of the Simeck cryptographic family and its resistance to well-known cryptographic attacks, including differential and linear cryptanalysis. The researchers found that Simeck shows promising potential as a practical solution that effectively balances security and efficiency, making it well-suited for its intended purposes. The performance evaluation compares Simeck with other lightweight BCs, showing its competitive nature and superior resource efficiency compared to alternative lightweight BCs. The Simeck software is effective for hardware implementations in resource-constrained environments such as IoT devices and embedded systems with restricted power, memory, and computation capabilities.

Beaulieu *et al.* [15] presented the SIMON and SPECK families of BCs, specifically created for resource-limited scenarios such as IoT devices, RFID tags, and sensor networks. The ciphers provide excellent security with simplicity and flexibility. SIMON offers several key lengths and block sizes, whereas SPECK utilizes additions and rotations for software implementations. Both ciphers are designed for hardware efficiency and are resilient against common cryptographic attacks such as differential and linear cryptanalysis. They provide competitive performance attributes, making them appropriate for minimal gate counts in hardware and efficient code in software. SIMON and SPECK are cryptographic primitives designed for devices with limited resources, making them ideal for a wide range of applications in lightweight cryptography because of their adaptability in block and key sizes.

Zhang *et al.* [16] developed the RECTANGLE BC, a lightweight cryptographic algorithm tailored for systems with constrained power, memory, and computational capabilities, such as IoT devices. The cipher has a bit-slice design, performing operations on individual bits instead of bytes, resulting in the advantages of parallelism and fast hardware and software executions. The cipher's details, such as block size, key size, number of rounds, S-boxes, and permutation layers, are deliberated about. The authors conducted a comprehensive analysis of RECTANGLE's security, subjecting it to common cryptographic attacks and assessing its resilience against potential vulnerabilities. The cipher's performance is evaluated and contrasted with various systems to showcase its adaptability and effectiveness. The RECTANGLE BC is a lightweight cryptography solution that offers security against known threats and efficiency when used on various platforms.

Borghoff *et al.* [17] PRINCE is a BC specifically created for use in pervasive computing situations that need low-latency applications. It provides a cryptographic solution with distinctive features for fast device encryption without compromising security. PRINCE is ideal for applications related to embedded electronic devices, smart appliances, and wearable technology. The experimental results and evaluations emphasize its efficiency in terms of speed and resource requirements. The article examines PRINCE's security aspects, focusing on vulnerabilities and potential attack scenarios. PRINCE fills a void in the cryptographic ecosystem by focusing on low latency, a specific need that is not met by many traditional ciphers.

Yao *et al.* [18] presented an LED cipher in this paper, which uses a dimming-down technique and includes open-source threshold characteristics. The LED threshold design utilized a finite-state machine to aid with data and state changes. Using the LED threshold allows for more chip area allocation and a higher operating frequency compared to an unprotected LED cipher. The threshold value (T) is used to prevent side-channel resistance attacks. Ayachi *et al.* [19] provide a scholarly analysis of the data encryption framework used in network-on-chip (NoC)-based LED ciphers. The network interface (NI) is designed for NoC architecture, with or without encryption. This paper analyzes the chip area of secure NI designs and investigates possible improvements. Tiwari *et al.* [20] describe the box-based active number calculation in the BC. Mixed integer linear programming (MILP) is utilized to identify the active substitution box (S-box). The Active S-Box computation assesses the LED, KLEIN, and AES ciphers' appropriateness by considering the number of rounds. Execution timing analysis is performed for each round, considering the activation of specific S-Boxes. Sliman *et al.* [21] introduce a series of BC tailored for IoT applications. These ciphers are known for their exceptionally low weight, making them ideal for IoT devices with limited resources. The paper examines different strategies, including involutive, bit-slice, additional-rotation XOR (ARX), and hybrid procedures, for creating BCs. The ultra-lightweight method (ULM) technique is distinguished by its capacity to reduce the number of clock cycles and total memory use when compared to other BCs.

Kaur and Singh [22] analyze the cryptographic techniques used in IoT applications in their research. An in-depth study is performed on symmetric-key and public-key ciphers. This review examines the performance metrics, benefits, and limitations of symmetric key and public-key ciphers. Salman *et al.* [23] conduct a thorough analysis of the AES with customized modifications designed for IoT applications. This paper investigates the modifications to the circular processes of the AES, with a specific emphasis on the shift rows and mixed columns transformations. The study suggested that the modified round in the AES reduces the overall number of rounds from 10 to six. Ghayoula *et al.* [24] extensively explain how they implemented the SIMON-128 BC on the Artix-7 FPGA. The SIMON-128 cryptographic algorithm utilizes the Feistel round structure, comprising a data route unit, a key scheduling unit, and a control unit. The design consumes 72 milliwatts (mW) and utilizes around 2% of the chip's slices. Ibrahim *et al.* [25] conducted a detailed analysis of lightweight block ciphers (LBC), emphasizing their implementation on FPGA architectures. This paper investigates various lightweight cryptographic algorithms such as AES, LED, extended tiny encryption algorithm (XTEA), PRESENT, and KLEIN. All performance metrics of the LBCs, such as area, throughput, frequency, and efficiency, are analyzed. Tsantikidou and Sklavos [26] examine the use of low-power Bluetooth controllers (LBCs) in IoT applications in the healthcare sector. The authors primarily concentrate on analyzing the hardware limitations linked to LBCs in this specific field. The IoT is thoroughly analyzed in relation to healthcare. This paper offers an in-depth examination of various cryptographic methods, such as BCs, stream ciphers, hash functions, and authentication protocols. Each method is thoroughly analyzed, taking into account its benefits, drawbacks, and performance attributes. Zhang *et al.* [27] explain the low bitrate coding (LBC) approach called GFRX, which is designed for IoT nodes. The GFRX algorithm is a hybrid integration of the Feistel and ARX techniques. GFRX can handle data sizes of 64, 96, and 128 bits and supports key sizes of 96, 128, and 256 bits. The security analysis includes numerous linear and structural investigations that have been reported.

Kabir *et al.* [28] examines prevalent dangers, attacks, and vulnerabilities in IoT devices and their extensive utilization. The text examines the difficulties presented by IoT devices and proposes effective solutions. The paper presupposes that readers have knowledge of IoT devices and associated technology. The document offers an overview of IoT device layers, identifies typical dangers, examines the difficulties in mitigating these threats, and proposes future actions to improve safety, security, and privacy. Future research could investigate zero-trust architectures, security-enhanced architectural layers, machine and deep learning models, cloud and edge computing solutions, and security as a universal concern.

3. PROPOSED SYSTEM MODEL

Guo *et al.* [29] initially proposed the LED cipher in 2012. It is known for its efficient utilization of resources on chips, particularly in comparison to other BCs when implemented on tiny hardware. The LED cipher exhibits similarities to the AES cipher in terms of encryption and decryption processes, with the exception of the key generation process. This study focuses on the modification of the LED cipher by introducing alterations to the round process, aiming to enhance both latency and throughput. The proposed model has been specifically engineered to possess a key size of 64/128 bits, which corresponds to a data size of 64 bits. The cipher structures of the proposed MLED-64 and MLED-128 are depicted in Figures 1 and 2, respectively. The MLED-64/128 cipher executes an add-round key (ARK) or XOR operation by utilizing a 64-bit plain text (P) and a 64-bit key (K). The ARK output persistently executed step procedures in order to produce the 64-bit cipher text (C). The MLED-64 and MLED-128 algorithms require 8 and 12 computational steps, respectively, to successfully execute encryption or decryption operations. The MLED-64 employs a uniform 64-bit key (K) for its ARK operations. In contrast, the MLED-128 algorithm partitions the 128-bit key into two distinct components, namely K1 and K2. The K1 and K2 variables, respectively, store the first [63:0] bits and the subsequent [127:64] bits, which are utilized for both encryption and decryption processes. The MLED-128 employs a 64-bit K1 key during the initial 8 steps, after which it transitions to the K2 key for the subsequent 4 steps (9 to 12).

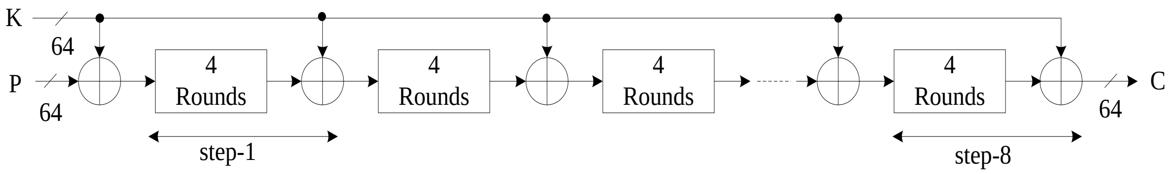

Figure 1. Modified LED-64 cipher architecture

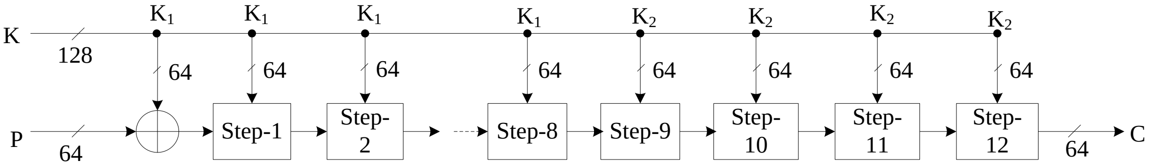

Figure 2. Modified LED-128 cipher architecture

Figure 3 depicts the operational process of the improved LED-64/128 cipher architecture in a single phase. The operational process consists of four rounds in each phase, which are subsequently followed by the add round key (ARK) operation. Each iteration consists of four distinct operations, namely the add constant module (ACM), sub cells module (SCM), shift row module (SRM), and mixed columns module (MCM). The round numbers (RNi) are defined by the user for each round. The value of RNi ranges from 0 to 7 for the MLED-64 cipher and from 0 to 11 for the MLED-128 cipher. The output of each round serves as the input for the subsequent round, and the output of Round 3 is subjected to an XOR operation with the associated key (K) in order to produce the output of a single-step operation. The subsequent section provides a detailed analysis of the circular operation of each individual module.

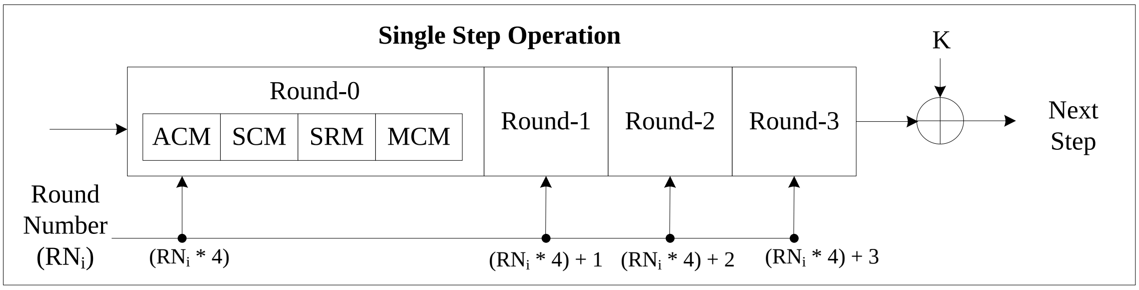

Figure 3. Single-step operation of modified LED-64/128 cipher architecture

This study examines the utilization of round constants in the ACM modules. Table 1 presents the tabulated round constants of the MLED-64/128 encryption. The MLED-64 cipher considers round numbers ranging from 0 to 31, whereas the MLED-128 cipher considers round values ranging from 0 to 47. The majority of current LED methodologies utilize six bits that undergo a left-shift operation and are afterwards substituted with a novel round constant value during the ACM procedure. The aforementioned procedure requires an additional four clock cycles per round in order to execute the left shifting operation, followed by the replacement of the previous value with a new one. The proposed methodology reduces the number of clock cycles required for each operation round in ACM by four. The S-Box is utilized by SCM to execute subcell operations. The SCM module receives a 64-bit value from the ACM, which is organized in a 4×4 matrix. It then proceeds to conduct the substitution operation on each row of the matrix. Table 2 presents the tabulated S-Box utilized in the SCM for the Modified LED-64/128 Cipher. The S-Box is characterized by a 4-bit input (in) and a corresponding 4-bit output (out). The input values consisting of 4 bits are substituted with corresponding S-Box values, which serve as the output. The aforementioned procedure is iterated a total of 16 times in order to produce the 64-bit SCM output.

Table 1. Round constants of modified LED-64/128 cipher

Rounds	Round constants
0-23	01, 03, 07, 0F, 1F, 3E, 3D, 3B, 37, 2F, 1E, 3C, 39, 33, 27, 0E, 1D, 3A, 35, 2B, 16, 2C, 18, 30
24-47	21, 02, 05, 0B, 17, 2E, 1C, 38, 31, 23, 06, 0D, 1B, 36, 2D, 1A, 34, 29, 12, 24, 08, 11, 22, 04

Table 2. S-Box used in SubCells module for modified LED-64/128 cipher

In	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Out	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

The SRM receives a 4×4 matrix of 64-bit SCM data. It then proceeds to conduct a row-wise shifting operation to the left, as illustrated in Figure 4. The operations of no-shifting, one-time left-shifting, two-time left-shifting, and three-time left-shifting are executed on the first, second, third, and fourth rows of the SRM, respectively.

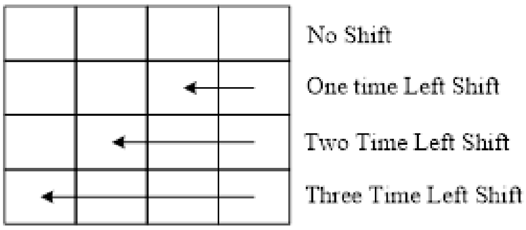

Figure 4. SRM operation

The MCM receives data in a 64-bit SRM format and does column-wise mixing operations. This study examines the two distinct 4 x 4 matrices utilized in the AES cipher [11] for the MCM (mix columns) and inverse MCM operations. The left-hand side (M) and right-hand side (M-1) of the 4x4 matrix are utilized for MCM and inverse matrix chain multiplication, respectively. These components are denoted in (1).

$$M = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \text{ and } M^{-1} = \begin{bmatrix} E & B & D & 9 \\ 9 & E & B & D \\ D & 9 & E & B \\ B & D & 9 & E \end{bmatrix} \tag{1}$$

The output of the 64-bit SRM is partitioned into four columns, and each column is subjected to multiplication with a M array iteratively until the last column is reached. The column data undergoes updates through multiplications with the M array, resulting in a 64-bit MCM output. The output of a single-step operation is generated by performing an XOR operation between the 64-bit MCM data and its matching key.

The decryption or deciphering process of the MLED-64/128 involves the reversal of the encryption operation. The architecture of the improved LED-128 decipher is depicted in Figure 5. The decryption procedure of MLED-128 involves the utilization of a 64-bit K2 key for the initial four phases, followed by the application of a K1 key for the subsequent eight steps (from step 4 to step 12). The output of step 12 is subjected to an XOR operation with the K1 key in order to produce the resulting 64-bit plaintext.

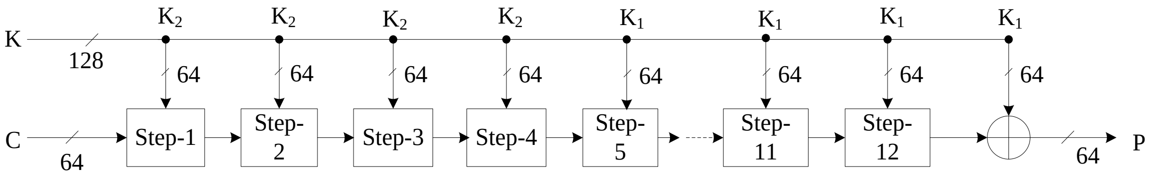

Figure 5. Modified LED-128 decipher architecture

Figure 6 depicts the inverse single-step execution of the improved LED-64/128 decipher architecture. Following each inverse step operation, there is an ARK and four rounds of operation. Each phase of the study includes the implementation of the inverse maximum causal model (IMCM), the inverse structural regression model (ISRM), the inverse structural causal model (ISCM), and the ACM. The IMCM algorithm employs an M-1 matrix for the purpose of performing matrix multiplication. The ISRM executes the process of left shifting in reverse order on a row-by-row basis. The ISCM uses an identical S-Box for cryptographic operations. However, the output value is regarded as the new input, and conversely, the previous input is treated as the output. The operation of the ACM remains consistent with the cipher process during the decoding procedure.

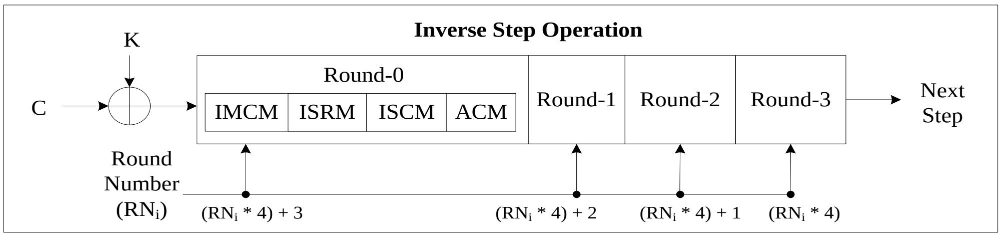

Figure 6. Inverse single-step operation of modified LED-64/128 decipher architecture

4. RESULTS AND DISCUSSION

This section presents a discussion on the simulation and synthesized findings of the improved LED-64/128 module. The LED-64/128 module has been adapted and executed on an Artix-7 FPGA utilizing Verilog-HDL within the Xilinx ISE framework. By using the Xilinx Tool to do place and route operations, we can find out the chip area (slices), maximum operating frequency, and total power of the redesigned LED-64/128 module and its sub-modules. The xilinx power analyzer (XPA) tool is used to find out the total power after the place and route procedures have been run. Performance measures such as latency, throughput, and efficiency are seen in the case of MLED-64/128 and its corresponding sub-modules. The simulation outcomes of the redesigned LED-64 and LED-128 modules are depicted in Figures 7 and 8, respectively. The functioning of the modified LED-64/128 module is initiated by activating the global clock (clk) through an asynchronous reset (rst) with an active low signal. The input data (inp) is specified as 64 bits, and the input key is specified as either 64 or 128 bits. The MLED-64 and MLED-128 cipher modules require 7.5 and 12.5 clock cycles (CC), respectively, in order to generate the encrypted output, also known as cipher_out. On the other hand, it is worth noting that the MLED-64 and MLED-128 decipher modules require 8.5 and 12.5 CC, respectively, in order to produce the decrypted output (decipher_out). The simulation results presented above demonstrate that both the MLED-64 and MLED-128 modules require the 64-bit original input and decrypted output values to be identical.

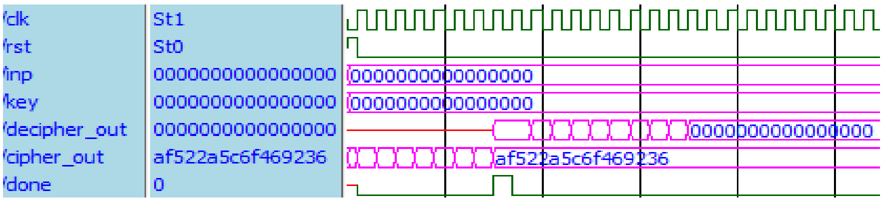

Figure 7. Simulation results of modified LED-64 module

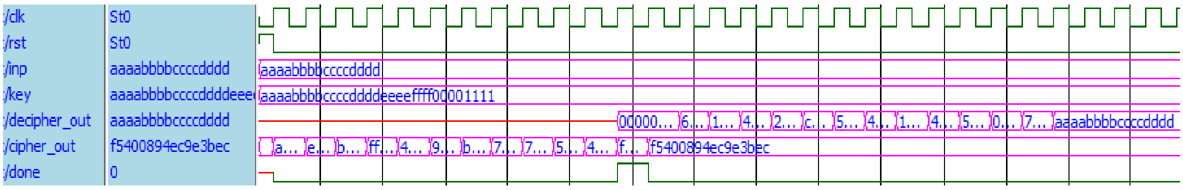
:/clk St0
:/rst St0
:/inp aaaabbbbccccdddd
:/key aaaabbbbccccddddeeee
:/decipher_out aaaabbbbccccdddd
:/cipher_out f5400894ec9e3bec
:/done 0
aaaabbbbccccdddd
aaaabbbbccccddddeeeeffff00001111
00000... 6... 1... 4... 2... c... 5... 4... 1... 4... 5... 0... 7... aaaabbbbccccdddd
a... e... b... ff... 4... 9... b... 7... 7... 5... 4... f... f5400894ec9e3bec

Figure 8. Simulation results of modified LED-128 module

Table 3 presents the tabulated data on the resource use of MLED-64/128 and its sub-modules on the Artix-7 FPGA. The encryption module known as MLED-64 (Enc) employs slices of 576 and runs at a maximum frequency of 160.136 MHz. It consumes a total power of 445 mW on the Artix-7 FPGA. The decryption module, known as MLED-64 Dec, employs 576 slices and runs at a frequency of 126.197 MHz, while requiring a total power of 862 mW. The MLED-64 encryption-decryption (ED) module employs 1157 slices and functions at a frequency of 125.186 MHz, with a total power consumption of 1.006 W. The measurement of delay is determined based on the number of CC. The determination of throughput or data rate, measured in megabits per second (Mbps), on an FPGA device relies on factors such as latency, data size, and the frequency achieved. The MLED-64 encryption module achieves a delay of 7.5 CC, a throughput of 1366.5 Mbps, and a hardware efficiency of 2.373 Mbps per slice. The decryption module MLED-64 demonstrates a latency of 8.5 CC, a throughput of 950.2 Mbps, and a hardware efficiency of 1.649 Mbps per slice. The MLED-64 ED module requires 16 computational cycles (CC) to perform encryption and decryption tasks. It delivers a throughput of 500.75 Mbps with an efficiency of 0.433 Mbps per computational slice.

The encryption module known as MLED-128 employs a total of 832 slices and functions at a maximum frequency of 160.136 MHz while consuming a total power of 563 mW. The decryption module, known as MLED-128, employs 832 slices and functions at a frequency of 125.387 MHz while requiring a total power of 1.087 W. The MLED-128 ED module employs the segments of 1669 and functions at a frequency of 125.004 MHz while consuming a total power of 1.446 W. The MLED-128 encryption module achieves a delay of 12.5 CC, a throughput of 819.89 Mbps, and a hardware efficiency of 0.986 Mbps per slice. The decryption module MLED-128 demonstrates a latency of 12.5 CC, a throughput of 641.99 Mbps, and a hardware efficiency of 0.772 Mbps per slice. The MLED-128 ED module exhibits a latency of 25 CC and attains a throughput of 320.12 Mbps, demonstrating an efficiency of 0.192 Mbps per slice. Figure 9 displays the graphical depiction of the resource use for the MLED-64/128 module. The Artix-7 FPGA demonstrates an approximate chip area utilization of 1% for both the MLED-64/128 and its sub-modules.

The performance evaluation of the suggested MLED ciphers in contrast to the existing LED ciphers on the FPGA platform has been presented in Table 4. The data/key size, FPGA type, chip area (slices), highest frequency the device can achieve, latency, throughput (in Mbps), and hardware efficiency (in Mbps per slice) are the resource characteristics for performance comparison. The LED cipher, as described in reference [30], utilizes a key size of 64/128 bits and is implemented on an Artix-7 FPGA. The MLED-64, as suggested, exhibits a noteworthy enhancement in latency by 97.07%, throughput by 94.4%, and efficiency by 52.38% compared to the LED-64 [31]. The MLED-128 cipher, as suggested, exhibits a notable enhancement in latency, with a 96.74% improvement, as well as a substantial increase in throughput by 94.14%. Additionally, when compared to the LED-128 encryption, as mentioned in [32], its efficiency is 29% higher. The LED cipher, as described in reference [14], utilizes a 128-bit key and is implemented on an Artix-7 FPGA. When compared to the LED-128 encryption, the MLED-128 cipher as described shows a significant improvement in latency, with a gain of 53.7%, as well as an increase in throughput of 21.57% [33]. The LED cipher, as described in reference [34], utilizes a 64/128-bit key and is implemented on a Spartan-6 FPGA.

Table 3. Resource utilization of modified LED-64/128 module on Artix-7

Resources	MLED-64 Enc	MLED-64 Dec	MLED-64 ED	MLED-128 Enc	MLED-128 Dec	MLED-128 ED
Slices	576	576	1157	832	832	1669
Max. frequency (MHz)	160.136	126.197	125.186	160.136	125.387	125.044
Total Power (mW)	445	862	1006	563	1087	1446
Latency (CC)	7.5	8.5	16	12.5	12.5	25
Throughput (Mbps)	1366.5	950.2	500.75	819.89	641.99	320.12
Efficiency (Mbps/Slices)	2.373	1.649	0.433	0.986	0.772	0.192

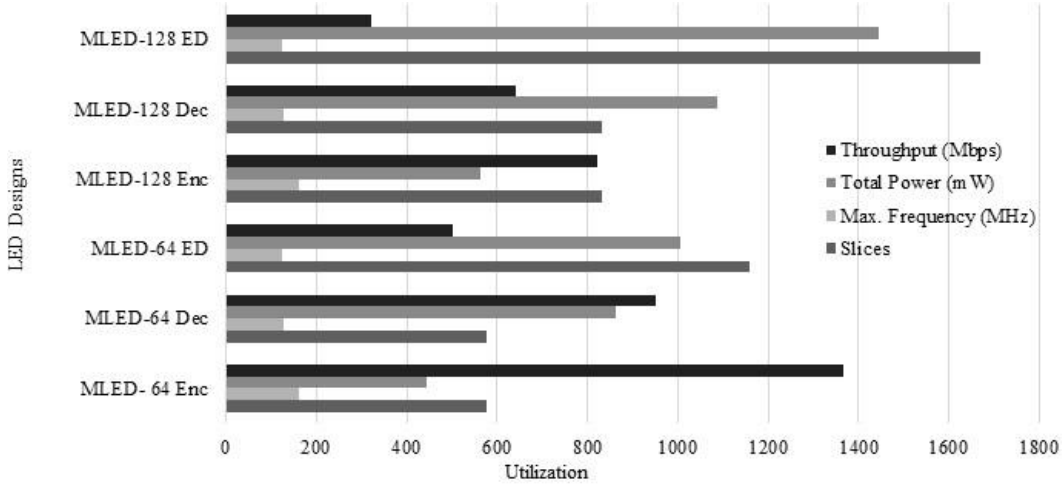

Figure 9. Graphical representation of performance metrics of modified LED-64/128 design

The MLED-64, as proposed, demonstrates a reduction in area overhead by 83.8%, an improvement in frequency by 90.1%, a decrease in latency by 77.27%, an increase in throughput by 26.06%, and an enhancement in efficiency by 88.03% as compared to the LED-64. The MLED-128, as described, demonstrates a significant reduction in area overhead by 78.1%, a notable improvement in frequency by 93.2%, a substantial decrease in latency by 74.4%, a moderate increase in throughput by 17.06%, and a substantial enhancement in efficiency by 80.62% as compared to the LED-128. The LED cipher, as described in reference, utilizes a 128-bit key and is implemented on a Spartan-6 FPGA. The MLED-128 cipher, as suggested, exhibits a notable enhancement in latency, with an improvement of 93.5% compared to the LED-128 cipher [35]. Additionally, it demonstrates a substantial increase in throughput, with a boost of 94.56%. Additionally, when compared to the LED-128 cipher, the MLED-128 cipher's efficiency is significantly higher by 80.73%.

The performance evaluation of the proposed LED designs is presented in Table 5, where a comparison is made with existing BC techniques. The MLED-128, as presented, exhibits a reduction in area overhead of 52.48%. Furthermore, it demonstrates an improvement in latency by 60.9%, throughput by 6.09%, and efficiency by 49.5% compared to SIMON-128. On the other hand, the MLED-128, as suggested, demonstrates a reduction in area overhead by 58.68%, an improvement in latency by 75.9%, an increase in throughput by 22.67%, and an enhancement in efficiency by 68.05% when compared to the SPECK-128 [36]. The MLED-128, as suggested, demonstrates enhancements in many performance metrics when compared to the SPECK-128 [37]. Specifically, it exhibits a 15.7% improvement in operating frequency, a 68.75% reduction in latency, a 73.53% increase in throughput, and a 44.92% boost in efficiency. The MLED-128, as suggested, demonstrates enhancements in operating frequency by 6.95%, throughput by 5.71%, and efficiency by 39.14% when compared to the RECTANGLE-128 [38]. When compared to the PRESENT-128 algorithm [18], the MLED-128 shows a big improvement in latency (91.8% less), throughput (87.9% more), and efficiency (50%) [39]. The MLED-128 algorithm, as suggested, demonstrates a notable enhancement in latency by 90.23%, throughput by 90.12%, and efficiency by 65.51% when compared to the XTEA-128 algorithm [40].

Table 4. Performance comparison of the proposed MLED ciphers with existing LED ciphers

Cipher Design	LED	LED	LED	LED	LED	LED	MLED-64	MLED-128
Data size	64	64	64	64	64	64	64	64
Key size	64	128	128	64	128	128	64	128
FPGA	Artix-7	Artix-7	Artix-7	Spartan-6	Spartan-6	Spartan-6	Artix-7	Artix-7
Slices	63	69	404	3556	3800	229	576	832
Frequency (MHz)	284	286	357	15.88	10.64	133.76	160.136	160.136
Latency (CC)	256	384	27	33	49	192	7.5	12.5
Throughput (Mbps)	71.21	47.75	643	1010	680	44.59	1366.5	819.89
Efficiency (Mbps/Slice)	1.13	0.7	1.59	0.284	0.191	0.19	2.373	0.986

Table 5. Performance comparison of the proposed LED designs with existing BC approaches

Cipher design	SIMON	SPECK	SPECK	RECTANGLE	PRESENT	XTEA	MLED-128
Data size	64	64	64	64	64	64	64
Key size	128	128	128	128	128	128	128
FPGA	Spartan-3	Spartan-3	Spartan-3	Spartan-3	Virtex-6	Artix-7	Artix-7
Slices	1751	2014	399	483	201	316	832
Frequency (MHz)	344	191	135	149	211	264	160.136
Latency (CC)	32	52	40	NA	136	128	12.5
Throughput (Mbps)	770	634	217	773	99.13	80.43	819.89
Efficiency (Mbps/Slice)	0.497	0.315	0.543	0.6	0.493	0.34	0.986

5. CONCLUSION

The MLED is a cipher-decipher system designed on the Artix-7 FPGA platform to prioritize high efficiency and minimal latency. The algorithm being discussed uses 8-step operations for MLED-64 and 12-step operations for MLED-128 for encryption and decryption. Each phase involves four sequential rounds in the operating procedure, followed by an add-round key operation. The circular process has been modified to improve latency in relation to clock cycles. The simulation and synthesized results are fully executed. The MLED-64/128 module utilizes 1% of the chip area (slices) and operates at a frequency of 125 MHz on the Artix-7 FPGA. The MLED-64 encryption uses 7.5 bits and achieves a throughput of 1366.5 Mbps, resulting in an efficiency of 2.373 Mbps per slice. The MLED-128 encryption uses 12.5 units and achieves a throughput of 819.89 Mbps, with an efficiency of 0.986 Mbps per slice. The MLED-64/128 cipher exhibits advantageous traits such as reduced latency and good throughput. These characteristics indicate that the encryption is suitable for use in applications that need little resources. The study compares the proposed approach with existing LED and other BC algorithms on the FPGA platform, showing significant improvements in performance limitations. The circular operation of the MLED-64/128 design is optimized to maximize the operating frequency on the FPGA platform and enhance performance results.

REFERENCES

[1] I. Bhardwaj, A. Kumar, and M. Bansal, "A review on lightweight cryptography algorithms for data security and authentication in IoTs," in *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, IEEE, Sep. 2017, pp. 504–509, doi: 10.1109/ISPCC.2017.8269731.

[2] M. Cazorla, K. Marquet, and M. Minier, "Survey and benchmark of lightweight block ciphers for wireless sensor networks," in *2013 International Conference on Security and Cryptography (SECRYPT)*, 2013, pp. 1–6.

[3] B. J. Mohd, T. Hayajneh, and A. V. Vasilakos, "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues," *Journal of Network and Computer Applications*, vol. 58, pp. 73–93, Dec. 2015, doi: 10.1016/j.jnca.2015.09.001.

[4] E. R. Naru, H. Saini, and M. Sharma, "A recent review on lightweight cryptography in IoT," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, Feb. 2017, pp. 887–890, doi: 10.1109/I-SMAC.2017.8058307.

[5] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, "A review of lightweight block ciphers," *Journal Cryptogr Eng*, vol. 8, no. 2, pp. 141–184, Jun. 2018, doi: 10.1007/s13389-017-0160-y.

[6] S. Surendran, A. Nassef, and B. D. Beheshti, "A survey of cryptographic algorithms for IoT devices," in *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, IEEE, May 2018, pp. 1–8, doi: 10.1109/LISAT.2018.8378034.

[7] A. Sevin and A. A. O. Mohammed, "A survey on software implementation of lightweight block ciphers for IoT devices," *Journal Ambient Intell Humaniz Computer*, vol. 14, no. 3, pp. 1801–1815, Mar. 2023, doi: 10.1007/s12652-021-03395-3.

[8] Nayancy, S. Dutta, and S. Chakraborty, "A survey on implementation of lightweight block ciphers for resource constraints devices," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 5, pp. 1377–1398, Jul. 2022, doi: 10.1080/09720502.2020.1766764.

[9] L. Xu, J. Guo, J. Cui, and M. Li, "Key-recovery attacks on LED-like block ciphers," *Tsinghua Science Technol*, vol. 24, no. 5, pp. 585–595, Oct. 2019, doi: 10.26599/TST.2018.9010130.

[10] M. S. Naik, D. K. Sreekantha, and K. V. S. S. S. S. Sairam, "Comparative study of block ciphers implementation for resource-constrained devices (Review)," *Известия высших учебных заведений. Радиоэлектроника*, Dec. 2023, doi: 10.20535/S0021347023050011.

[11] A. Bogdanov *et al.*, "PRESENT: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 450–466, doi: 10.1007/978-3-540-74735-2_31.

[12] H. M. Al-Saadi and I. S. Alshawi, "Efficient and secure hybrid chaotic key generation for light encryption device block cipher," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 31, no. 2, p. 1032, Aug. 2023, doi: 10.11591/ijeecs.v31.i2.pp1032-1040.

[13] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: An ultra-lightweight blockcipher," 2011, pp. 342–357, doi: 10.1007/978-3-642-23951-9_23.

[14] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, "The simeck family of lightweight block ciphers," *In International workshop on cryptographic hardware and embedded systems*, 2015, pp. 307–329, doi: 10.1007/978-3-662-48324-4_16.

[15] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proceedings of the 52nd Annual Design Automation Conference*, New York, NY, USA: ACM, Jun. 2015, pp. 1–6, doi: 10.1145/2744769.2747946.

[16] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms," *Science China Information Sciences*, vol. 58, no. 12, pp. 1–15, Dec. 2015, doi: 10.1007/s11432-015-5459-7.

[17] J. Borghoff *et al.*, "PRINCE – a low-latency block cipher for pervasive computing applications," *Cryptology ePrint Archive,* 2012, pp. 208–225, doi: 10.1007/978-3-642-34961-4_14.

[18] Y. Yao, M. Yang, P. Kiaei, and P. Schaumont, "Dimming down LED: An open-source threshold implementation on light encryption device (LED) block cipher," *arXiv preprint arXiv:2108.12079*, 2021.

[19] R. Ayachi, A. Mhaouch, and A. B. Abdelali, "Lightweight cryptography for network-on-chip data encryption," *Security and Communication Networks*, vol. 2021, pp. 1–10, May 2021, doi: 10.1155/2021/9943713.

[20] V. Tiwari, N. Jampala, A. N. Tentu, and A. Saxena, "Towards finding active number of s-boxes in block ciphers using mixed integer linear programming," *Informatica*, vol. 45, no. 6, Oct. 2021, doi: 10.31449/inf.v45i6.3427.

[21] L. Sliman, T. Omrani, Z. Tari, A. E. Samhat, and R. Rhouma, "Towards an ultra lightweight block ciphers for internet of things," *Journal of Information Security and Applications*, vol. 61, p. 102897, Sep. 2021, doi: 10.1016/j.jisa.2021.102897.

[22] A. Kaur and G. Singh, "Encryption algorithms based on security in IoT," in *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, IEEE, Oct. 2021, pp. 482–486, doi: 10.1109/ISPCC53510.2021.9609495.

[23] R. S. Salman, A. K. Farhan, and A. Shakir, "Lightweight modifications in the advanced encryption standard (AES) for IoT applications: a comparative survey," in *2022 International Conference on Computer Science and Software Engineering (CSASE)*, IEEE, Mar. 2022, pp. 325–330, doi: 10.1109/CSASE51777.2022.9759828.

[24] R. Ghayoula, J. Fattahi, A. Smida, I. El Gmati, E. Pricop, and M. Ziadia, "FPGA implementation of SIMON-128 cryptographic algorithm using Artix-7," *In 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI),* 2023.

[25] M. S. Ibrahim, Y. A. Abbas, and M. H. Ali, "The performance of various lightweight block ciphers FPGA architectures: A review," *Al-Iraqia Journal of Scientific Engineering Research*, vol. 1, no. 1, Sep. 2022, doi: 10.33193/IJSER.1.1.2022.43.

[26] K. Tsantikidou and N. Sklavos, "Hardware limitations of lightweight cryptographic designs for IoT in healthcare," *Cryptography*, vol. 6, no. 3, p. 45, Sep. 2022, doi: 10.3390/cryptography6030045.

[27] X. Zhang, S. Tang, T. Li, X. Li, and C. Wang, "GFRX: A new lightweight block cipher for resource-constrained IoT nodes," *Electronics (Basel)*, vol. 12, no. 2, p. 405, Jan. 2023, doi: 10.3390/electronics12020405.

[28] M. A. A. Kabir, W. Elmedany, and M. S. Sharif, "Securing IoT devices against emerging security threats: challenges and mitigation techniques," *Journal of Cyber Security Technology*, vol. 7, no. 4, pp. 199–223, 2023, doi: 10.1080/23742917.2023.2228053.

[29] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," *In Cryptographic Hardware and Embedded Systems–CHES 2011: 13th International Workshop, Nara, Japan, September 28–October 1*, 2011, pp. 326–341, doi: 10.1007/978-3-642-23951-9_22.

[30] S. Heron "Advanced encryption standard (AES)," Gaithersburg, MD, Nov. 2001, doi: 10.6028/NIST.FIPS.197.

[31] J. Yang, L. Li, Y. Guo, and X. Huang, "DULBC: A dynamic ultra-lightweight block cipher with high-throughput," *Integration*, vol. 87, pp. 221–230, Nov. 2022, doi: 10.1016/j.vlsi.2022.07.011.

[32] N. N. Anandakumar, T. Peyrin, and A. Poschmann, "A very compact FPGA implementation of LED and PHOTON," *In Progress in Cryptology--INDOCRYPT 2014: 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings 15,* 2014, pp. 304–321, doi: 10.1007/978-3-319-13039-2_18.

[33] S. P. Guruprasad and B. S. Chandrasekar, "An evaluation framework for security algorithms performance realization on FPGA," in *2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, IEEE, Feb. 2018, pp. 1–6, doi: 10.1109/ICCTAC.2018.8370396.

[34] B. Rashidi, "Flexible structures of lightweight block ciphers PRESENT, SIMON and LED," *IET Circuits, Devices and Systems*, vol. 14, no. 3, pp. 369–380, May 2020, doi: 10.1049/iet-cds.2019.0363.

[35] W. E. H. Youssef, A. Abdelli, F. Dridi, and M. Machhout, "Hardware implementation of secure lightweight cryptographic designs for IoT applications," *Security and Communication Networks*, vol. 2020, pp. 1–13, Nov. 2020, doi: 10.1155/2020/8860598.

[36] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "SIMON and SPECK: Block ciphers for the internet of things," *Cryptology ePrint Archive,* 2015, [Online]. Available: https://eprint.iacr.org/2015/585.

[37] A. Nemati, S. Feizi, A. Ahmadi, and V. A. Makki, "A low-cost and flexible FPGA implementation for SPECK block Cipher," in *2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, IEEE, Sep. 2015, pp. 42–47, doi: 10.1109/ISCISC.2015.7387896.

[38] S. Feizi, A. Nemati, A. Ahmadi, and V. A. Makki, "A high-speed FPGA implementation of a bit-slice ultra-lightweight block cipher, RECTANGLE," in *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, IEEE, Oct. 2015, pp. 206–211, doi: 10.1109/ICCKE.2015.7365828.

[39] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Lightweight hardware architectures for the present cipher in FPGA," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2544–2555, Sep. 2017, doi: 10.1109/TCSI.2017.2686783.

[40] R. Anusha and V. V. Devi Shastrimath, "LCBC-XTEA: High throughput lightweight cryptographic block cipher model for low-cost rfid systems," *In Cybernetics and Automation Control Theory Methods in Intelligent Algorithms: Proceedings of 8th Computer Science On-line Conference*, 2019, pp. 185–196, doi: 10.1007/978-3-030-19813-8-20.

BIOGRAPHIES OF AUTHORS

Mahendra Shridhar Naik is working as a senior assistant professor at New Horizon College of Engineering in Bengaluru. He is currently pursuing a Ph.D. in crytography. He has more than 10 years of teaching experience. He received his M.Tech in Digital Electronics and Communication Engineering from Visvesvaraya Technological University (VTU), Belagavi, in 2013. His areas of interest include VLSI, IoT, wireless sensor networks, and data science. He can be contacted at email: mahendrasnaik@gmail.com.

Desai Karanam Sreekantha has been serving as Professor in the Dept. of Computer Science and Engineering at NMAM Institute of Technology, NITTE, and India since Nov. 2014 and was awarded Ph.D. from Symbiosis International University, Pune, in 2014. He has secured Second Rank at Gulbarga University in B.Sc. (Electronics) degree examinations and National Merit Scholarship. He has 23 years of teaching experience and 6 years of industry experience in the TATA group. He authored about 25 Scopus-indexed papers, i.e., one book, 13 book chapters, and 25 papers in international journals, and he also presented 30 research papers at international/national conferences. He published two Indian patents. He was currently guiding four Ph.D. students. He can be contacted at email: sreekantha@nitte.edu.in.

Kanduri VSSSS Sairam working as a Professor, E&CE Department and also IEEE student branch Counsellor in NMAMIT, NITTE. He obtained his BE (ECE) from Karnataka University Dharwad in 1996, M.Tech (Industrial Electronics) from SJCE, Mysore University Mysore, 1998, and Ph.D. (ECE) (Optical Communications) JNTUH, Hyderabad in 2013. He has 25 years of experience in Teaching and Research too. He published 54 papers at International, National Conferences and Workshops. He is guiding four Ph.D. students and 40 M.Tech Projects and BE Projects. His research areas are optical communications, optical networks, and wireless communication. He can be contacted at email: drsairam@nitte.edu.in.