International Information and Engineering Technology Association
Advancing the World of Information and Engineering

International Journal of Safety and Security Engineering
Vol. 15, No. 9, September, 2025, pp. 1819-1825
Journal homepage: http://iieta.org/journals/ijsse

FPGA Implementation of SFN Lightweight Encryption Algorithm

Yasir A. Abbas[1*], Miaad H. Mahdi[2], Saad Al-Azawi[2]

[1] College of Engineering for Artificial Intelligence Technology, University of Diyala, Baqubah 32001, Iraq
[2] College of Engineering, University of Diyala, Baqubah 32001, Iraq

Corresponding Author Email: dr.yasiral-zubaidi@uodiyala.edu.iq

Copyright: ©2025 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.150906

Received: 3 August 2025
Revised: 15 September 2025
Accepted: 25 September 2025
Available online: 30 September 2025

Keywords:
cryptography, lightweight, substitution-permutation, Feistel network, IoT, hardware optimization, low-power design

ABSTRACT

Efficient cryptography algorithms and security systems are essential to ensure the security of the transmitted information. However, the IoT devices and sensors suffer from their limited processing capabilities and power constraints. Thus, in such cases, the traditional cryptographic algorithms will not be efficient methods to provide security for such devices. Therefore, lightweight block cipher algorithms have emerged as a solution to secure resource-constrained devices. This paper presents an efficient implementation of the Substitution-Permutation (SP) Network and Feistel Network (SFN) lightweight Block Cipher algorithm using a field programmable gate array (FPGA). The SFN algorithm emerged as an efficient and lightweight algorithm that represents a suitable choice to provide protection for IoT devices and sensors. The novelty of the proposed SFN architecture is represented by a low hardware utilization rate and the maintenance of high performance. The performance results show low power consumption while preserving a low utilization rate and high performance in comparison to similar lightweight block cipher architectures.

1. INTRODUCTION

The Internet of Things (IoT) is the environment through which many devices are connected with each other in order to gather and exchange data via the internet. The number of IoT devices is increasing in various fields such as healthcare, smart homes, and industrial automation to provide convenience and more efficiency. However, this increasing number of IoT devices is accompanied by security challenges because they have limited computational resources and do not have any security measures [1, 2].

The security of IoT devices is of utmost importance to guarantee the confidentiality, integrity, and availability of the data transmitted among the communicating devices. The IoT devices are vulnerable to a lot of attacks, including unauthorized access, denial-of-service, and data breach attacks, due to their heterogeneous nature, which has produced conflicting security protocols. As a consequence, designing efficient security techniques suitable for the unique constraints of IoT devices is very ticklish [3, 4].

Advanced traditional cryptographic algorithms cannot be used as a solution to provide security measures for resource-constrained devices. Consequently, lightweight encryption algorithms emerged as a solution to ensure security requirements for limited-resource devices [5, 6]. Lightweight algorithms consider that the IoT devices have low memory and limited computational capabilities. Thus, it is a fundamental reason to use lightweight algorithms in resource-limited devices to provide a secure environment for data communication. On the one hand, such algorithms preserve battery life, computing power, and memory [7-9]. Also, some resource-constrained devices suffer from processing capabilities or software stacks to ensure efficient software execution for such algorithms. Thus, a hardware implementation of lightweight algorithms is essential to provide protection for the transmitted data for such devices. As such, Field programmable gate array (FPGA) platforms represent a suitable environment for designing, implementing, and testing lightweight algorithms. Their programmable nature helps in optimizing cryptographic algorithms to suit different applications in the IoT. Developers can benefit from FPGAs by achieving high throughput and low power consumption to ensure that lightweight cryptographic algorithms are efficient and secure [10-12].

Many cryptography algorithms employ either a substitution permutation network (SPN) or a Feistel structure (FN). However, an efficient lightweight algorithm that utilizes both SPN and FN networks to carry out the encryption and decryption functions has been introduced in the study [13]. Three concepts are gathered to constitute a lightweight block cipher called SFN. The first concept was to solve the limitation of dissimilarity of encryption and decryption processes of the SPN network by utilizing the related properties of the linear and nonlinear components. Then, the second, they made both operations of cryptography, encryption, and decryption work as an FN structure. Finally, a MixRows layer is added in the SPN part.

In this paper, a new iterative architecture is designed for the SFN lightweight algorithm using an FPGA platform to provide protection for the data transmitted by embedded devices of

IoT. Specifically, for devices with limited processing capabilities or devices with no software stacks to execute security algorithms as software.

The rest of this paper is organized as follows: Section 2 presents a review of the related lightweight cryptography architectures. Section 3 introduces a description of the SFN algorithm. The proposed SFN architecture is introduced in Section 4. The performance evaluation of the proposed architecture and a comparison with other architectures are introduced in Section 5, and the conclusions are drawn in Section 6.

2. RELATED WORK

This section presents a review of the related hardware implementation of similar lightweight cryptography algorithms. In study [14], a high-speed, low-area FPGA architecture of the Advanced Encryption Standard (AES) was developed for cryptography applications. The most significant feature of the architecture is replacing the Look-Up Tables (LUTs) with combinational logic circuits. However, the main disadvantage is the increase in latency in specific stages due to registers for the sub-pipelining structure.

Two architectures (serial and iterative) were implemented on a 32-bit data path for the LED block cipher in study [15] tailored for resource-constrained devices. Hardware utilization is highly optimized by performing sequential operations. But the serial architecture had high latency compared to the iterative architecture.

The hardware architecture of the PRESENT block cipher algorithm in study [16] reduces area utilization by 42% compared to an iterative architecture developed by studies [17-19]. Moreover, the serial implementation achieved high throughput using a 16-bit data path when compared to other serial architectures. On the other hand, the serial architecture offers an increased latency of 155 clock cycles in comparison with the iterative architectures.

A serial hardware architecture is introduced in the study [20] for the PRINCE block cipher algorithm using a 16-bit data path to ensure efficient resource usage. In addition, this architecture used a Dynamic Linear Feedback Shift Register (DLFSR) to enhance its security against brute-force attacks. Nevertheless, the serial design produced higher latency in comparison with parallel approaches. As a result, the design is not suitable for real-time applications that require a high-speed encryption and decryption process.

The GIFT block cipher algorithm [21] was developed to address the key limitations of the PRESENT algorithm in terms of high hardware cost and susceptibility to linear cryptanalytic attacks. Although, fix-slicing technique is fastest in its execution time, it requires more memory and energy than other methods. In the same way, the LUT-based technique takes more time to execute.

In study [22], iterative and serial architectures for the Piccolo lightweight algorithm have been introduced. Although the serial architecture shows significant area optimization compared to the iterative architecture, it requires high latency to process one block of data.

Further, a Simple Hybrid Cipher (SHC) with 64-bit data and 128-bit key for the new lightweight block cipher algorithm is proposed in the study [23]. The S-Box of this block cipher algorithm is designed based on composite field arithmetic (CFA). However, the S-Box complexity requires longer development time and increases hardware costs.

An iterative architecture has been designed for the ANU-II lightweight block cipher in study [1] for encryption and decryption processes. The designed architecture achieved a high throughput of up to 3831.19 Mbps and an efficiency of 46.72 Mbps/slice using the Virtex-6 FPGA platform. Moreover, only 13 clock cycles are required to process one 64-bit block of data across 25 rounds.

A hardware implementation for the AES-128 algorithm is proposed in the study [24] tailored for the 5G communication devices. It achieved a high throughput of up to 28.16 Gbps with a low latency of 10 clock cycles only. However, the decryption architecture requires more hardware resources than the encryption architecture.

In study [25], two hardware-efficient architectures have been proposed for the LED cipher with 8 and 4 bits. The 8-bit design achieved a balance between performance and area. Yet, it requires 816 clock cycles to process one block of data.

In such circumstances, this paper introduces an efficient architecture for the SFN block cipher algorithm that was introduced in the study [13]. The proposed architecture considers the constraints of IOT devices in memory, hardware resources, and processing capabilities.

Many lightweight encryption algorithms were introduced to overcome the limited capabilities of small and mobile devices. The PRESENT algorithm was considered an efficient, low-hardware-cost, and compact block cipher algorithm. But the PRESENT has a shortcoming of inflexible key scheduling and low resistance to linear cryptography. Thus, the GIFT algorithm was developed to improve the security and reduce the power consumption of the PRESENT by improving the S-boxes and permutation stages. However, even though both algorithms preserve static architectural design, they do not have high flexibility to adapt to various hardware platforms. As such, the SFN algorithm considers both Substitution-Permutation and Feistel structures, offering an optimized solution for low hardware cost, low power consumption, and security performance. Such a high level of performance of the SFN makes it an optimum solution for IoT and real-time applications.

3. THE SFN ALGORITHM

The SFN algorithm merges the SPN and Feistel structures to provide a more efficient block cipher algorithm. The SFN algorithm structure is shown in Figure 1 for 64-bit data and 96-bit key [13]. The SFN primary key consists of 96 bits, of which the last 32-bit is utilized as control signals for the SPN and FN networks of the algorithm. When the SPN is used for Key expansion, the Feistel Network is used for encryption or decryption. If the control signal is zero, the SPN structure is used for the encryption or decryption, and the Feistel network is employed for Key expansion.

The lightweight SFN block cipher is implemented on an FPGA using a key optimization by using lookup tables to carry out the multiplication operations in the MixColumns layers instead of traditional arithmetic operations. Hardware complexity is reduced by avoiding costly arithmetic operations, and this key optimization technique significantly enhances speed.

The main technique employed to optimize FPGA resource usage was the look-up table. A look-up table was used to perform Mix-Column with comparator sets without utilizing multipliers, XORs, and irreducible polynomials operations, to obtain the results of any two matrix multiplication. High speed and low power consumption are provided as a result of using a comparator and reduced mathematical and logic operations [26].

The specification of the SFN algorithm consists 64-bit plaintext, a 96-bit key, and the number of rounds is 32. The 32-bit control key is used to switch between SP and Feistel networks. Each bit is used to determine if SP is employed for encryption or Key expansion, and in the same way for the Feistel structure.

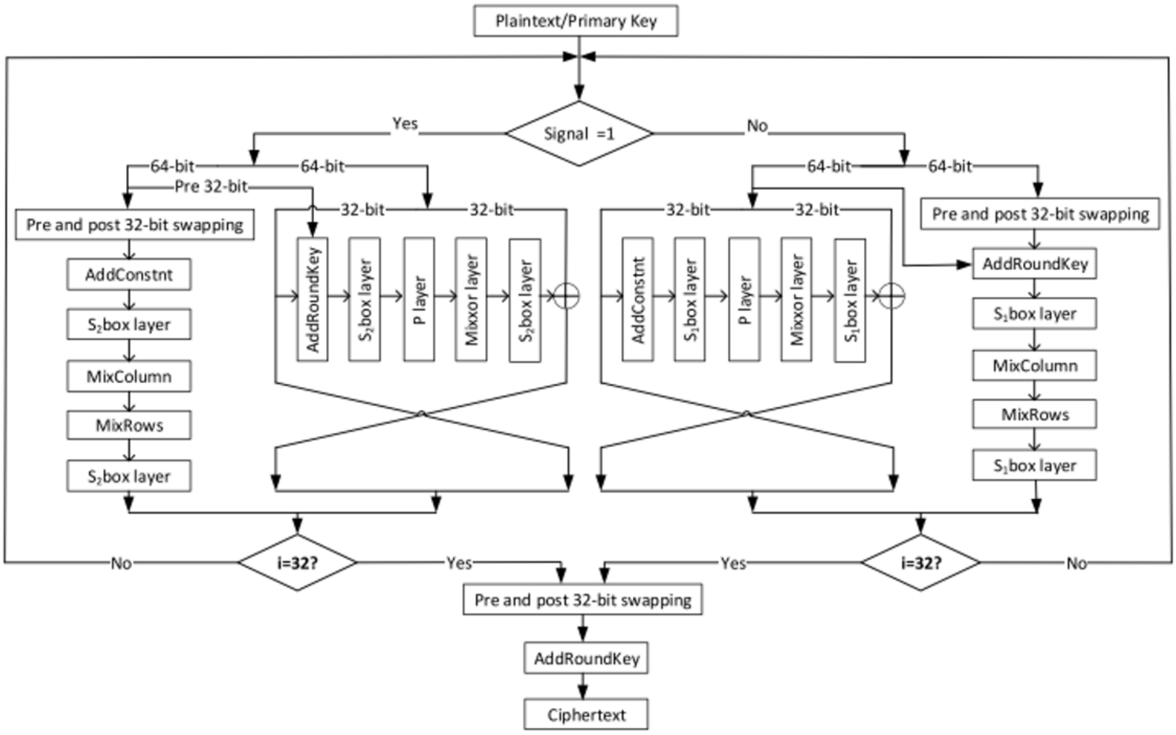

Figure 1. SFN block cipher algorithm [13]

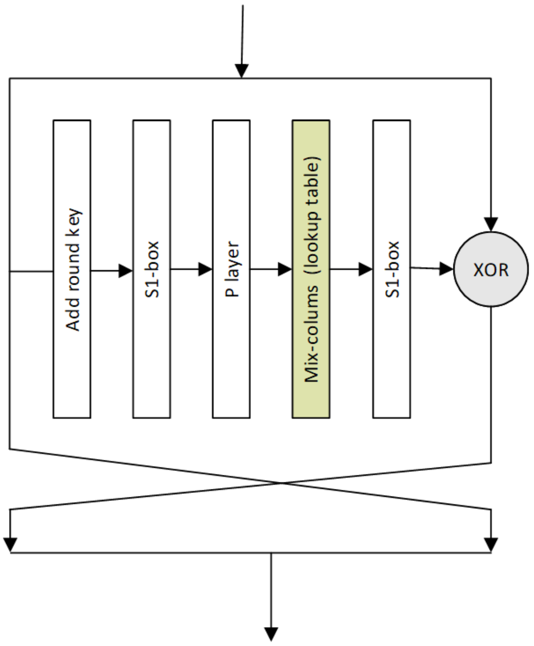

Figure 2. The SP network

X	0	1	2	3	4	5	6	7
$S_1[x]$	C	A	D	3	E	B	F	7
X	8	9	A	B	C	D	E	F
$S_1[x]$	8	9	1	5	0	2	4	6

Figure 3. S1-Box [13]

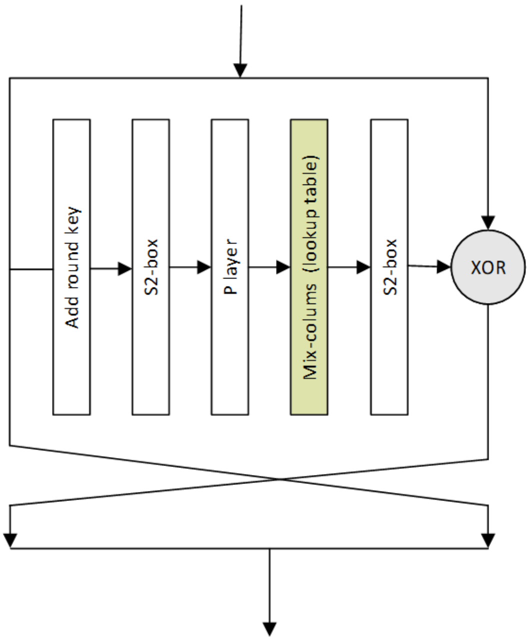

Figure 4. The Feistel network

X	0	1	2	3	4	5	6	7
$S_2[x]$	B	F	3	2	A	C	9	1
X	8	9	A	B	C	D	E	F
$S_2[x]$	6	7	8	0	E	5	D	4

Figure 5. S2-Box [13]

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P (*j*)	9	28	7	13	8	12	29	6	0	2	17	23	30	24	18	11
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P (*j*)	31	4	15	19	5	1	25	27	3	10	22	21	26	16	20	14

Figure 6. The P-layer [13]

The round function of the SP network includes AddRoundKey, which performs an XOR operation between the current state and the round key, as shown in Figure 2. Next, the state passes through a 4 × 4 S-box (S1-Box), which is illustrated in Figure 3. Then, the 4 × 4 Maximum Distance Separable (MDS) matrix is utilized to perform the linear diffusion layer. Another layer of diffusion (MixRows) is employed with the same MDS matrix. Finally, the S1-Box is executed on the state.

The first layer of the Feistel Network is the AddRoundKey, in which the round key is XORed with the state, as depicted in Figure 4. Next, the Substitution layer (S2-Box), Figure 5, is employed. Then, the diffusion is executed using the Bit permutation (P-layer) as described in Figure 6.

After the diffusion layer, the linear transformation layer (MixXors) using XOR operations is performed. Finally, the S2-Box layer is executed again on the state. Finally, the S2-Box layer is executed again on the state.

4. THE PROPOSED ARCHITECTURE

The iterative FPGA architecture of the SFN lightweight algorithm, as depicted in Figure 7, adopts a round-based layout where a 64-bit state goes through repeated operations across 32 iterations, controlled by a global clock signal. The datapath is organized to reuse functional units, leading to minimizing hardware overhead while preserving cryptographic strength. At each iteration, the state passes through sequential operations, including pre- and post-32-bit swapping, constant addition, nonlinear substitution layers (S1-Box and S2-Box), permutation, and mix-columns/mix-rows implemented via lookup tables. A comparator depends on the last 32 bits of the key to select the appropriate processing path, while to guarantee diffusion and confusion, an XOR logic combines the round key with the state. This way ensures an optimal equilibrium between throughput and resource efficiency, making the SFN algorithm well-suited for FPGA-based lightweight cryptographic applications in constrained environments such as IoT and wireless sensor networks.

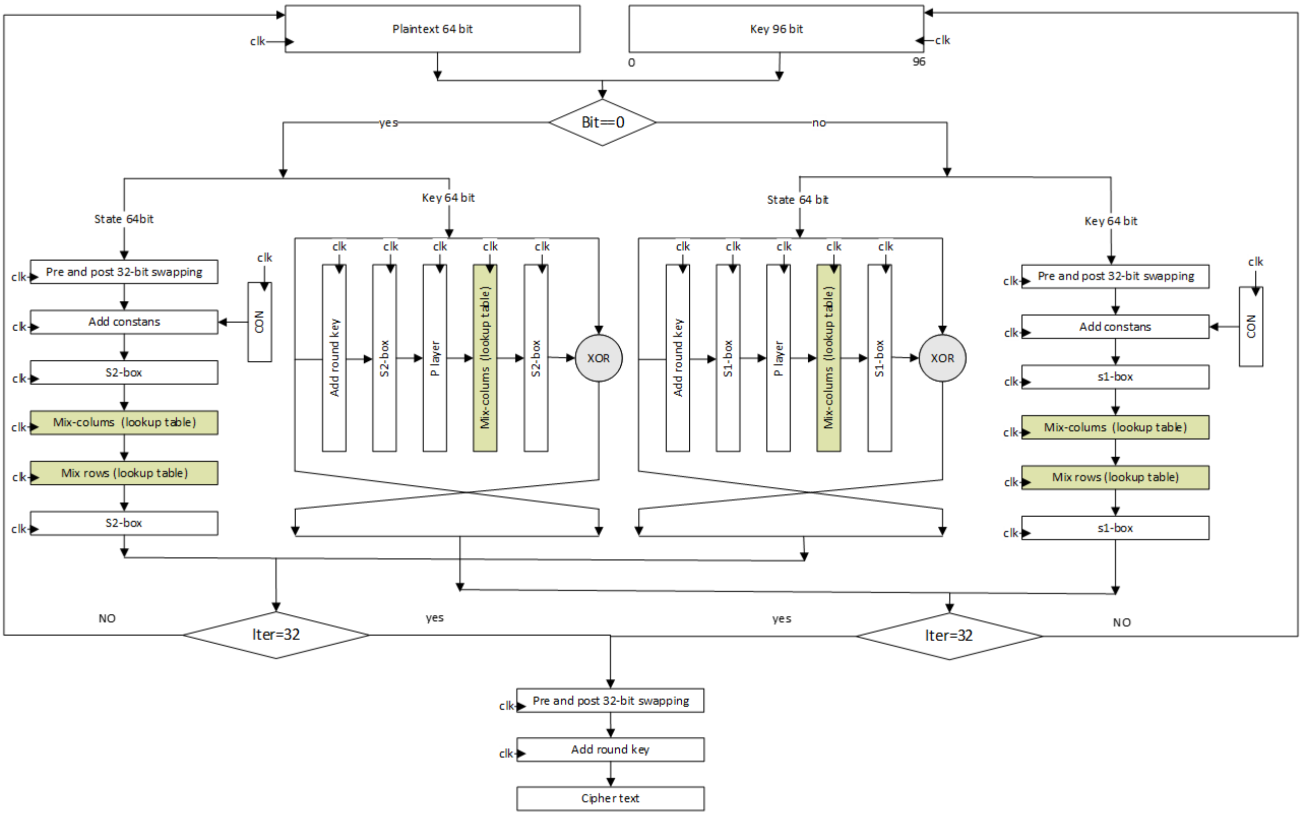

Figure 7. FPGA architecture of the SFN algorithm

5. PERFORMANCE RESULTS AND DISCUSSION

In this work, Xilinx ISE Design Suite 14.7 was used to design, simulate, and implement the architecture of the SFN lightweight algorithm. The design was described in VHDL and synthesized using the ISE environment, where functional simulation was carried out to verify correctness before hardware deployment. The target platform was the Xilinx Spartan-3 FPGA platform, chosen for its balance of logic resources, power efficiency, and suitability for lightweight cryptographic applications. Parameter configurations such as clock frequency, synthesis constraints, and area utilization were optimized to achieve a trade-off between speed and hardware resource consumption. The design was then subjected to the place-and-route process to evaluate timing, area occupancy, and estimated power consumption. A set of test benches was developed to validate encryption and decryption operations under various input conditions, ensuring functional accuracy and robustness. The hardware verification on the Spartan-3 board was conducted using the ISim 14.7 version P.58f simulation tools along with in-circuit testing, providing reliable results for analyzing the performance of the SFN lightweight cipher on the FPGA.

The performance of the proposed SFN architecture is evaluated using a Spartan-3 FPGA (xc6slx150t-3-fgg676) platform. The performance results and a comparison with similar architectures are illustrated in Table 1. As shown in Table 1, with a data size of 64 bits and a key size of 96 bits, the SFN achieves a maximum frequency of 477.897 MHz—the highest among all Spartan-3-based architectures. Such a high clock rate introduced an impressive throughput of 955.794 Mbps, while maintaining a latency of only 32 cycles. Also, the area utilization was 960 flip-flops, 601 LUTs, and 279 slices, resulting in a competitive efficiency of 3.425 Mbps per slice. Compared to other lightweight cipher architectures, such as ANU and Piccolo, the performance of the proposed SFN architecture outperforms them, obviously. For example, ANU (D4) and Piccolo both achieved lower throughput values of 671.03 Mbps and 168.9 Mbps with lower efficiencies of 2.467 and 0.495 Mbps/slice, respectively. Thus, the higher efficiency of the SFN architecture indicates a better use of FPGA resources, making it suitable for embedded applications where power and area are critical constraints.

In addition, SFN's performance becomes more impressive when it is compared with resource-conservative architectures such as the iterative and serial implementations of the PRESENT algorithm on the Spartan-III (XC3S50-5) platform [16]. While those architectures minimize area (e.g., 168 slices for the serial version), their throughput was 76.5 Mbps, and an efficiency of 0.45 Mbps/slice is considerably lower. Even the iterative version, which improves throughput to 306.58 Mbps, while it achieved an efficiency of 1.06 Mbps/slice, only. In this context, the SFN block cipher provides a superior trade-off between speed and resource consumption, making it ideal for scenarios requiring high-performance encryption with moderate hardware cost. The results clearly indicate that the SFN cipher achieves better performance compared to competitive architectures for the same FPGA family (Spartan-3), showcasing its strength for low-power, high-speed applications.

In comparison with other lightweight ciphers such as Piccolo and PRESENT, the SFN algorithm demonstrates superior performance in terms of both latency (32 cycles vs. 125 cycles for ANU and 31–155 cycles for PRESENT/Piccolo) and throughput. For example, the Piccolo cipher on Spartan-3 reaches only 168.9 Mbps at 81.82 MHz, while SFN attains nearly 6× higher throughput on the same family of FPGA boards. Similarly, PRESENT achieves modest area efficiency but falls behind SFN in terms of both maximum frequency and data processing speed, indicating SFN’s better suitability for time-critical IoT environments.

Potential improvements include extending the SFN architecture with pipeline and parallelization techniques to further reduce latency and enhance throughput, particularly for high-speed communication systems. Additionally, future work could investigate reconfigurable architectures that enable dynamic adjustment of parameters such as key size and block size, thereby improving flexibility across diverse application domains. Addressing these aspects would not only validate the robustness of the SFN cipher under more demanding environments but also enhance its applicability to broader cryptographic scenarios, including cloud security, mobile networks, and embedded systems with stringent performance and energy constraints.

In summary, the hardware implementation of the SFN lightweight cipher demonstrates a balanced and optimized performance profile when compared to other cryptographic implementations listed in Table 1.

Table 1. Comparison between the proposed SFN architecture and the round-based implementation of similar lightweight block cipher algorithms

FPGA Platform	Cipher/Architecture	Data Size (bit)	Key Size (bit)	Flip Flops	No. of LUT	No. of Slices	Latency	Maximum Frequency (MHz)	Throughput (Fmax)/(Mbps)	Efficiency Mbps/Slice
Spartan-3 xc6slx150t-3-fgg676	**The proposed SFN architecture**	64	96	960	601	279	32	477.897	955.794	3.425
Spartan-3 xc3s700an- 5fgg484	ANU [27]	64	128	210	460	265	125	243.253	124.54	0.46996
Spartan-3 (xc3s700an-5fgg484)	ANU(D4) [28]	64	128	199	513	272	25	262.123	671.03	2.467
Spartan-3 XC3S50pq208-5	Piccolo [22]	64	128	207	757	397	31	81.82	168.9	0.495
Virtex Ultrascale FPGA	AES [29]	128	128	1826	5352	-	11	175	2036.36	-
Spartan-6 /(XC6SLX45T-3) LUT	LED [30]	64	128	133	300	134	16	209.428	837.72	6.26
Spartan-3 XC3S50pq208-5	Piccolo [22]	64	128	207	757	397	31	81.82	168.9	0.495
Spartan-3/XC3S50-5 [iterative]	PRESENT [16]	64	128	199	555	289	31	148.5	306.58	1.06
Spartan-3/XC3S50-5 [serial]	PRESENT [16]	64	80	171	490	168	155	185.28	76.5	0.45

6. CONCLUSIONS

The implementation of the SFN lightweight block cipher has demonstrated superior performance and resource efficiency when compared to a range of other lightweight cryptographic algorithms. The achieved results, with a maximum frequency of 477.897 MHz, throughput of 955.794 Mbps, and efficiency of 3.425 Mbps/slice, clearly establish the SFN as a balanced and scalable solution for embedded applications requiring high security and low power consumption.

The performance of the proposed SFN architecture demonstrates superior trade-offs between area, speed, and throughput when compared against existing implementations such as the ANU, Piccolo, PRESENT, and LED. The SFN cipher consistently demonstrates superior trade-offs between area, speed, and throughput, particularly within the same FPGA family. While some newer FPGA platforms like Spartan-6 or Virtex-Ultrascale offer higher raw throughput or efficiency, they come at a high cost in terms of area and power, factors that are critical in IoT environments. SFN's ability to perform competitively on a lower-cost FPGA like Spartan-3 illustrates its design efficiency and practical applicability for real-time embedded systems that lack extensive computational resources. Moreover, its architecture, which integrates SPN and FN functionalities based on control key signals, offers additional flexibility and optimized encryption/decryption paths. Therefore, SFN stands out not only for its technical metrics but also for its alignment with the growing needs of secure, low-power, and high-speed IoT devices.

REFERENCES

[1] Yousif, N.H., Abbas, Y.A., Ali, M.H. (2022). Lightweight ANU-II block cipher on field programmable gate array. International Journal of Electrical and Computer Engineering, 12(3): 2194-2205. https://doi.org/10.11591/ijece.v12i3.pp2194-2205

[2] Kumar, S., Kumar, D., Dangi, R., Choudhary, G., Dragoni, N., You, I. (2024). A review of lightweight security and privacy for resource-constrained IoT devices. Computers, Materials and Continua, 78(1): 31-63. https://doi.org/10.32604/cmc.2023.047084

[3] Suryateja, P.S., Rao, K.V. (2024). A survey on lightweight cryptographic algorithms in IoT. Cybernetics and Information Technologies, 24(1): 21-34. https://doi.org/10.2478/cait-2024-0002

[4] Mahdi, M.H., Ibrahim, I.A. (2022). Routing protocols for hybrid wireless networks: A brief review. Indonesian Journal of Electrical Engineering and Computer Science, 27(2): 842-848. https://doi.org/10.11591/ijeecs.v27.i2.pp842-848

[5] Pandey, S., Bhushan, B. (2024). Recent Lightweight cryptography (LWC) based security advances for resource-constrained IoT networks. Wireless Networks, 30(4): 2987-3026. https://doi.org/10.1007/s11276-024-03714-4

[6] Mahdi, M.H., Ibrahim, I.A. (2023). Enhancing the security of quality of service-oriented distributed routing protocol for hybrid wireless network. Indonesian Journal of Electrical Engineering and Computer Science, 30(1): 121-128. https://doi.org/10.11591/ijeecs.v30.i1.pp121-128

[7] Sreehari, B., Sankar, V., Lopez, R.S., Vaishnav, K.S., Stuart, C.M. (2023). A review on FPGA implementation of lightweight cryptography for wireless sensor network. In 2023 International Conference on Power, Instrumentation, Control and Computing (PICC), Thrissur, India, pp. 1-6. https://doi.org/10.1109/PICC57976.2023.10142503

[8] Singh, S., Sharma, P.K., Moon, S.Y., Park, J.H. (2024). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. Journal of Ambient Intelligence and Humanized Computing, 15(2): 1625-1642. https://doi.org/10.1007/s12652-017-0494-4

[9] Radhakrishnan, I., Jadon, S., Honnavalli, P.B. (2024). Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices. Sensors, 24(12): 4008. https://doi.org/10.3390/s24124008

[10] Abbas, Y.A., Jidin, R., Jamil, N., Z'aba, M.R., Rusli, M.E., Tariq, B. (2014). Implementation of PRINCE algorithm in FPGA. In Proceedings of the 6th International Conference on Information Technology and Multimedia, Putrajaya, Malaysia, pp. 1-4. https://doi.org/10.1109/ICIMU.2014.7066593

[11] Ibrahim, M.S., Abbas, Y.A., Ali, M.H. (2022). The performance of various lightweight block ciphers FPGA architectures: A review. Al-Iraqia Journal for Scientific Engineering Research, 1(1): 124-129. https://doi.org/10.33193/IJSER.1.1.2022.43

[12] Abbas, Y., Jidin, R., Jamil, N., Z'aba, M. (2016). Reusable data-path architecture for encryption-then-authentication on FPGA. International Review on Computers and Software (IRECOS), 11(1): 56-63. https://doi.org/10.15866/irecos.v11i1.8367

[13] Li, L., Liu, B., Zhou, Y., Zou, Y. (2018). SFN: A new lightweight block cipher. Microprocessors and Microsystems, 60: 138-150. https://doi.org/10.1016/j.micpro.2018.04.009

[14] Kumar, T.M., Reddy, K.S., Rinaldi, S., Parameshachari, B.D., Arunachalam, K. (2021). A low area high speed FPGA implementation of AES architecture for cryptography application. Electronics, 10(16): 2023. https://doi.org/10.3390/electronics10162023

[15] Mhaouch, A., Ayadi, W., Ridha, S., Issa, K., Abdelali, A. B., Machhout, M. (2024). An efficient hardware implementation of LED lightweight block cipher. In 2024 IEEE 7th International Conference on Advanced Technologies, Signal and Image Processing (ATSIP), Sousse, Tunisia, pp. 312-316. https://doi.org/10.1109/ATSIP62566.2024.10638985

[16] Mhaouch, A., Elhamzi, W., Abdelali, A.B., Atri, M. (2022). Efficient serial architecture for present block cipher. In 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Hammamet, Tunisia, pp. 45-49. https://doi.org/10.1109/SETIT54465.2022.9875564

[17] Kavun, E.B., Yalcin, T. (2011). RAM-based ultra-lightweight FPGA implementation of PRESENT. In 2011 International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico, pp. 280-285. https://doi.org/10.1109/ReConFig.2011.74

[18] Lara-Nino, C.A., Morales-Sandoval, M., Diaz-Perez, A. (2016). Novel FPGA-based low-cost hardware architecture for the PRESENT block cipher. In 2016 Euromicro Conference on Digital System Design (DSD), Limassol, Cyprus, pp. 646-650. https://doi.org/10.1109/DSD.2016.46

[19] Kumari, A.S., Mandi, M.V. (2019). Implementation of present cipher on FPGA for IoT applications. IJERT, 8(8): 26-29.

[20] Teja, P.R., Sasamal, T.N. (2023). Implementation of efficient serial architecture for prince block cipher with enhanced security. In 2023 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, pp. 1-6. https://doi.org/110.1109/ICSCAN58655.2023.10395246

[21] Syed, I., Nazish, M., Sultan, I., Banday, M.T. (2022). Implementation techniques for GIFT block cypher: A real-time performance comparison. In 2022 Smart Technologies, Communication and Robotics (STCR), Sathyamangalam, India, pp. 1-5. https://doi.org/10.1109/STCR55312.2022.10009581

[22] Mhaouch, A., Elhamzi, W., Atri, M. (2020). Lightweight hardware architectures for the piccolo block cipher in FPGA. In 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, pp. 1-4. https://doi.org/10.1109/ATSIP49331.2020.9231586

[23] Kumar, S., Kumar, D., Lamkuche, H., Sharma, V.S., Alkahtani, H.K., Elsadig, M., Bivi, M.A. (2024). SHC: 8-bit compact and efficient s-box structure for lightweight cryptography. IEEE Access, 12: 39430-39449. https://doi.org/10.1109/ACCESS.2024.3372388

[24] Visconti, P., Velazquez, R., Capoccia, S., De Fazio, R. (2021). High-performance AES-128 algorithm implementation by FPGA-based SoC for 5G communications. International Journal of Electrical and Computer Engineering (IJECE), 11(5): 4221-4232. https://doi.org/10.11591/ijece.v11i5.pp4221-4232

[25] Mhaouch, A., Elhamzi, W., Abdelali, A.B., Atri, M. (2023). Efficient design for a hardware implementation of the LED block cipher. Journal Européen des Systèmes Automatisés, 56(5): 725-733. https://doi.org/10.18280/jesa.560502

[26] Abbas, Y.A., Jidin, R., Jamil, N., Z'aba, M.R., Al-Azawi, S. (2018). Small footprint mix-column serial for photon and LED lightweight cryptography. In 2018 International Conference on Advanced Science and Engineering (ICOASE), Duhok, Iraq, pp. 70-74. https://doi.org/10.1109/ICOASE.2018.8548802

[27] Dahiphale, V., Bansod, G., Zambare, A. (2019). Lightweight datapath implementation of ANU cipher for resource-constrained environments. In Intelligent Computing-Proceedings of the Computing Conference, London, United Kingdom, pp. 834-846. https://doi.org/10.1007/978-3-030-22868-2_57

[28] Dahiphale, V., Bansod, G., Zambare, A., Pisharoty, N. (2020). Design and implementation of various datapath architectures for the ANU lightweight cipher on an FPGA. Frontiers of Information Technology & Electronic Engineering, 21(4): 615-628. https://doi.org/10.1631/FITEE.1800681

[29] Prakashan, P., Gupta, H. (2024). A configurable AES implementation on FPGA for secure 5G systems. In 2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Kothamangalam, Kerala, India, pp. 1-4. https://doi.org/10.1109/RAICS61201.2024.10689863

[30] Naik, M.S., Sreekantha, D.K., Sairam, K.V. (2024). An efficient low-latency and high throughput LED cipher architecture for IoT security on a hardware platform. SN Computer Science, 5(7): 908. https://doi.org/10.1007/s42979-024-03275-5

NOMENCLATURE

AES	Advanced Encryption Standard
CFA	composite field arithmetic
DLFSR	Dynamic Linear Feedback Shift Register
FPGA	field programmable gate array
GIFT	Generalized Irregular Feistel Cipher
HDL	Hardware Description Language
LUT	Look-Up Table
SFN	Substitution-Permutation Network and Feistel Network (SFN)
SHC	Simple Hybrid Cipher
SP	Substitution-Permutation
SPN	substitution permutation network