Indonesian Journal of Electrical Engineering and Computer Science
Vol. 23, No. 3, September 2021, pp. 1674~1680
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v23.i3.pp1674-1680

Efficient hardware implementation for lightweight mCrypton algorithm using FPGA

Yasir Amer Abbas, Ahmed Salah Hameed, Safa Hazim Alwan, Maryam Adnan Fadel
Department of Computer Engineering, College of Engineering, University of Diyala, Baqubah, Iraq

Article Info

Article history:

Received Feb 26, 2021
Revised Jul 27, 2021
Accepted Aug 8, 2021

Keywords:

FPGA
Hardware architecture
Lightweight
mCrypton
VHDL

ABSTRACT

The lightweight cryptography is used for low available resources devices such as radio frequency identification (RFID) tags, internet of things (IoTs) and wireless sensor networks. In such case, the lightweight cryptographic algorithms should consider power consumption, design area, speed, and throughput. This paper presents a new architecture of mCrypton lightweight cryptographic algorithm which considers the above-mentioned conditions. Resource-shared structure is used to reduce the area of the new architecture. The proposed architecture is implemented using ISE Xilinx V14,5 and Spartan 3 FPGA platform. The simulation results introduced that the proposed design area is 375 of slices, up to 302 MHz operating frequency, a throughput of 646 Mbps, efficiency of 1.7 Mbps/slice and 0.089 Watt power consumption. Thus, the proposed architecture outperforms similar architectures in terms of area, speed, efficiency and throughput.

Corresponding Author:

Ahmed Salah Hameed
Department of Computer Engineering, University of Diyala
Diyala, Baquba, College of Engineering Branch P.O.BOX-1, Iraq
Email: ahmedhameed_eng@uodiyala.edu.iq

1. INTRODUCTION

Lightweight cryptography is used in limited hardware/software resources devices. Clear examples of low-resources devices are RFID tags, internet of things (IoTs) and wireless sensor networks. In low-resources environment, lightweight cryptographic algorithms should be built with a major focus on power consumption, size of hardware, and the cost of implementation [1]-[4]. Low-resources devices like RFID and sensors have a limitation in memory size, power source, and implementation area, to be fitted with applications in which they are equipped [5]-[8]. This limitation in resources makes the implementation of standard ciphers on, such as these devices, hard [9]. Security applications for lightweight cryptography offers high security services for the internet of things and cloud computing depends on authentication and confidentiality [10]-[13]. The hardware design for lightweight cryptography must optimize important constraint like area, power and latency.

The aim of this paper is to provide a hardware implementation for lightweight cryptographic algorithm that is suitable for small and mobile devices. An optimized implementation of all parts and techniques of the lightweight cryptographic algorithm mCrypton is presented. Different cryptographic processes will be fully implemented in VHDL using Xilinx ISE software, version 14.5 with all related simulations. The proposed implementation of mCrypton algorithm is designed with a small number of slices providing a high frequency, high throughput, and high efficiency for the proposed design when compared to previous works.

This research paper is organized as shown in; in Section 2 the related works are introduced. In Section 3 the mCrypton algorithm is briefly recalled. Subsequently, in Section 4 the proposed implementation is described, then the results are presented and compared to FPGA implementations for different block cipher algorithms. Finally, in Section 5 this paper is concluded.

2. RELATED WORKS

Many and different lightweight cryptographic algorithms have been previously proposed such as PRESENT [14], ICEBERG [15], and AES [16]. A group of existing optimized lightweight cryptographic schemes are discussed here. In [17], presented a reduced area implementation of an IP-core of PRINCE algorithm [17]. The input data of proposed block cipher can be encrypted with one cycle due to the use of parallel model design. The real time testing provided in [17] shows an efficient hardware design with low energy consumption.

In Soliman *et al.* [18] presented an optimized two versions of the AES algorithm in which a small and low power consumption implementation for security applications in IoT is provided. These designs use an iterative looping and pipelined architecture in developing the technique of implementing the AES-128 standard algorithm [18]. In [19] proposed reliable error detection architectures for two of famous Cryptographic algorithms (Simon and Speck). The proposed architectures have increased in the coverage of error detection and reduced in design complexity. The power, area, and delay time of the new implementation are acceptable and the design is efficient for low resources lightweight applications [19].

Mhaouch *et al*. [20] proposed an optimized version for Piccolo block cipher in FPGA. Two suggested designs iterative and serial architectures of Piccolo cipher are presented and showed a reduced area implementation related with improvement in speed compared to the standard implementation of the algorithm [20]. Abdullah *et al*. [21] suggested a new flexible architecture to implement PRINCE algorithm for high speed, small area, and low power design. The FPGA implementation is used to build a process of encryption with quantum cryptography protocol (BB84) in one clock cycle. The presented architectures could fit all basic cryptographic algorithms that use to build algorithms for applications like smart card and other portable devices [21].

3. MCRYPTON ALGORITHM

MCrypton is a 64-bit lightweight block cipher cryptographic algorithm presented in 2006 [22], [23]. Substitution permutation (SP) structure is used in design of mCrypton algorithm architecture. The algorithm is classified according to the key size in to mCrypton-64, mCrypton-96 and mCrypton-128. The proposed implementation is an architecture of 64-bit mCrypton with a key size of 64-bit. The overall view of the presented architecture is shown in Figure 1.

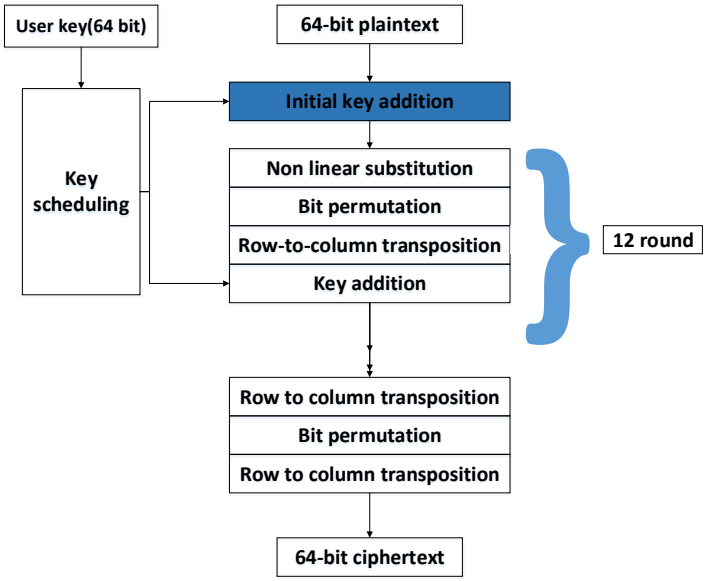

Figure 1. The overall view of the presented architecture

MCrypton and Crypton algorithms use a 4×4 array to represent an 8-byte data block [22]. The algorithm mainly has five different processes: the nonlinear substitution process, the bit permutation process, the row-to column transposition process, key scheduling process and key addition process. Twelve rounds of the five processes mentioned above are applied to the plaintext with a 64-bit initial key. The row-to column transposition process repeated twice and the bit permutation process repeated once before providing the 64-bit cipher text.

4. VHDL IMPLEMENTATION OF MCRYPTON ALGORITHM

The top module of the proposed design is shown in Figure 2. The design involves three input ports and one output port. Two of the input ports are used as interface ports to the plaintext 64-bit and the key 64-bit while the third input port is used as 1-bit enable to the system design. The output port is representing the 64-bit ciphertext. There is no need to use a large FPGA board since the total number of the pins that was used as input/output is 193 only.

In Figure 3, data flow for the designed hardware is shown. Each different process of mCrypton algorithm is built as standalone component and in which the plaintext and key is processed. Twelve repeated rounds of transformation are used to build the process of encryption. Each round of transformation will go through all the four stages of the algorithm [22].

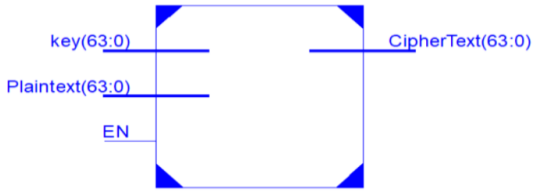

Figure 2. Top module of mCrypton RTL

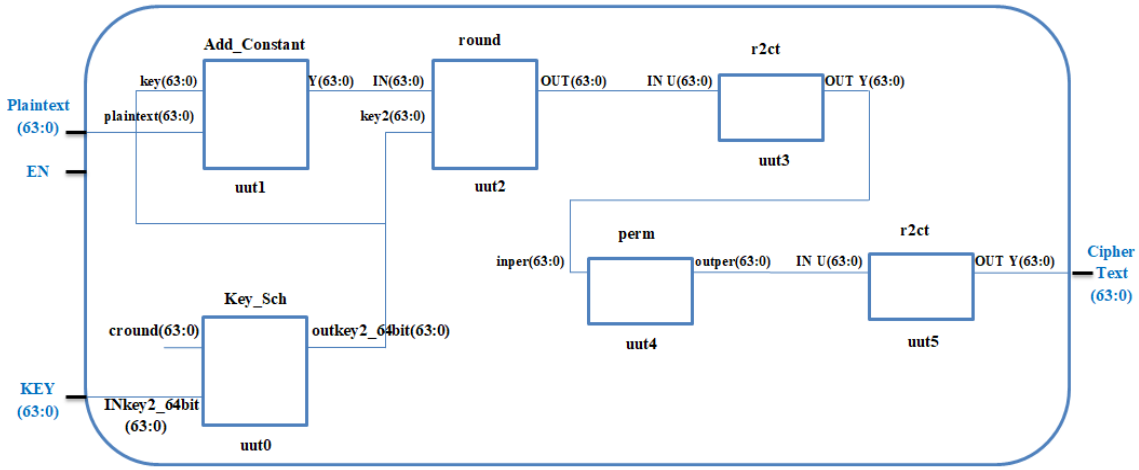

Figure 3. The data flow of the mCrypton

4.1. Nonlinear Substitution

In this block a 4×4 nipple array and four S-boxes of size 4-bit (S0, S1, S2, and S3) are used to do the process of nonlinear substitution as shown in (1)-(3).

$$S2 = S0\text{-}1 \text{ and } S3 = S1\text{-}1 \quad (1)$$

$$a = (a0, a1, a2, a3) \quad (2)$$

$$\gamma i(a) = (Si(a0), Si+1(a1), Si+2(a2), Si+3(a3)) \quad (3)$$

The substitution unit is designed with 64-bit for input and 64-bit for output. This component contains 64 LUT that is built with the ROM of size 16*4 to change value of S-boxes. Figure 4 shows the RTL of substitution component.

4.2. Permutation

To build a permutation process a hardwire structure is used rather than building it with a shift circuit that increase the area of the design. This component in very simple architecture using VHDL and the slice number is very small because it built from hardware wiring to transfer location of data and AND gate with constant. Figure 5 shows the number of the input/output buffer and AND gate that used to build a permutation operation in hardware.

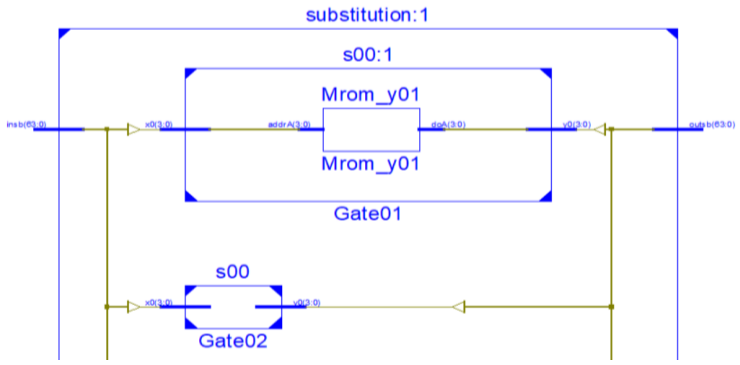

Figure 4. RTL substitution component of mCrypton

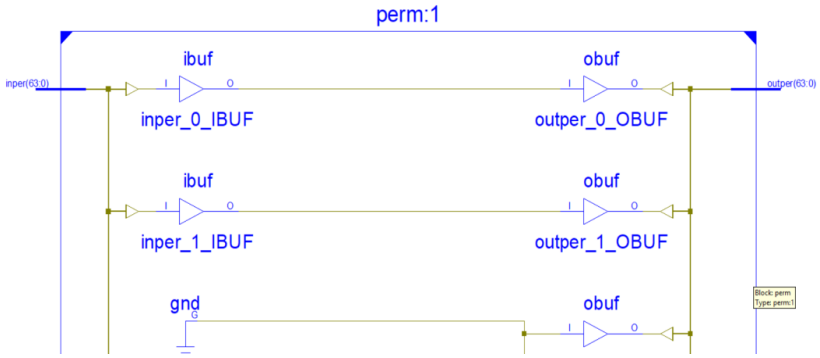

Figure 5. RTL permutation component of mCrypton

4.3. Row-to-column transposition

Moving the nipples with (i, j) location addresses into (j, i) location addresses can easily provide row-to-column transposition. This operation did not cost the hardware design because the hard wire is change the location of 4-bit from row to column.

4.4. Key scheduling

In mCrypton, the key scheduling algorithm involves two operations generate the round key operation using S-box and update the key variables with rotation [24], [25]. In the proposed architecture a RAM of constant key variables is used. The number of key variables is twelve which is equivalent to the round number of the transformations. The simple architecture is used to decrease hardware, in addition same component using in round is used is part. Figure 6 shows the S-Box and XOR operation that used in the first part of the key scheduling component.

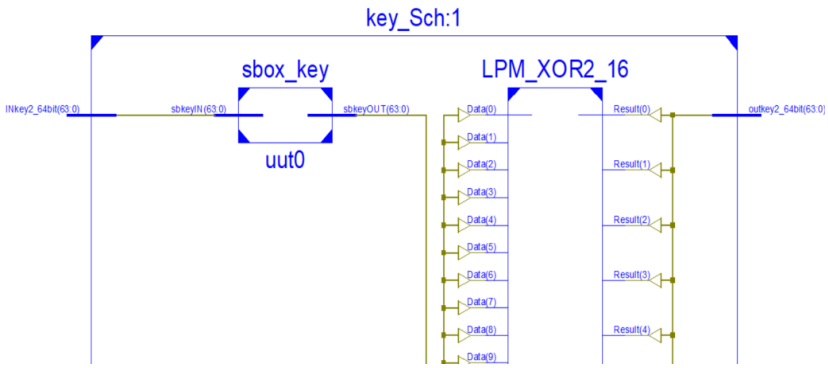

Figure 6. RTL key scheduling component of mCrypton

4.5. Key addition

It is the process of adding the result of an r2c process to the key provided by the key scheduling process. The process of addition is a simple bit to bit x-or logic operation.

5. SIMULATIONS AND RESULTS

Lightweight mCrypton algorithms were designed and tested using Xilinx ISE software, version 14.5 and all simulations have been done by ISim. The proposed algorithm has been designed with VHDL language. Using ISim, the VHDL codes were analyzed and synthesized, placed and routed in FPGA devices Spartan 3-xc3s1000-5fg320. Different performance metrics such as the area, throughput and power were computed.The low latency and low hardware implementation are the target of the design presented in this paper. The total slices 375, the mCrypton consist from add constant 64 slices, key scheduling 64 slices, the round that consists from (substitution, permutation, transfer and add constant) 96 slices, substitution has only been 32, finally the permutation and Row to Columns transfer in very small because it is hardware wire only.

The hardware implementation has been tested using ISim simulation software. A test vectors for the plaintext 64-bit and the key 64-bit enter the designed system as inputs. The processing of the inputs with the different components of the proposed design and producing the ciphertext is taking 30 clock cycles. Figure 7 shows the simulation results of the last round of data entered into the designed system and the last three components.

The proposed architecture has been proved to be efficient for working with high frequency and high throughput using a small number of slices. The slice number is reduced in shift operation by using LUT technique. LUT uses the change of locations between input and output data to achieve the shift operation. The shift operation that is built with LUT could be executed in one clock cycle. Using a small number of slices can reduce the cost of design and make the proposed design suitable to be used with RFID devices and IoT application.

A comparative of area (total number of slices), power (mWatt), and throughput (Mbps) were shown in Figures 8, 9, and 10 respectively. The proposed architecture results, show good throughput with small area and low power consumption as it is shown in Table 1. Results have been compared with different studies. The results of the proposed designed show a throughput of 646 Mbps and efficiency of 1.7 Mbps/slice with total power equal to 89 mWatt only.

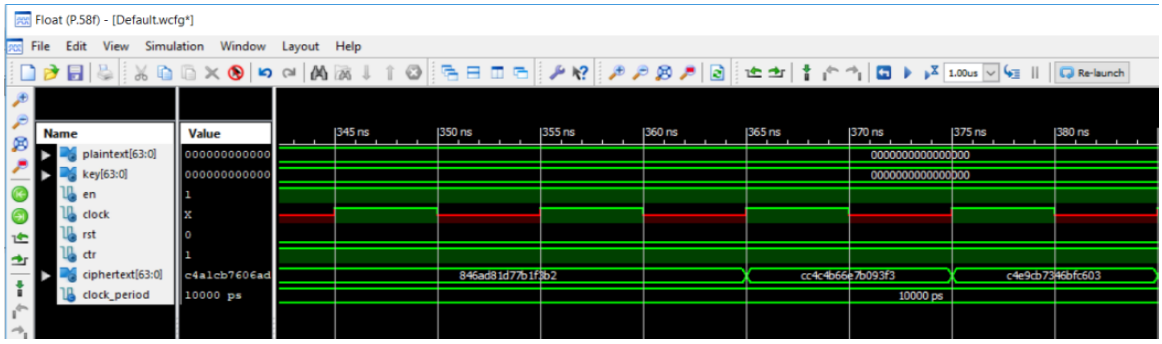

Figure 7. The test vector simulation of mCrypton

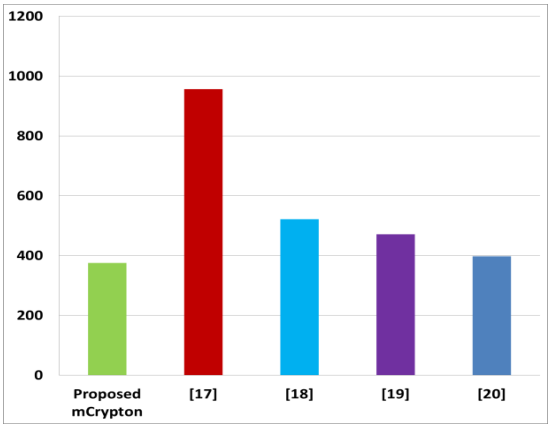

Figure 8. Total number of slices results

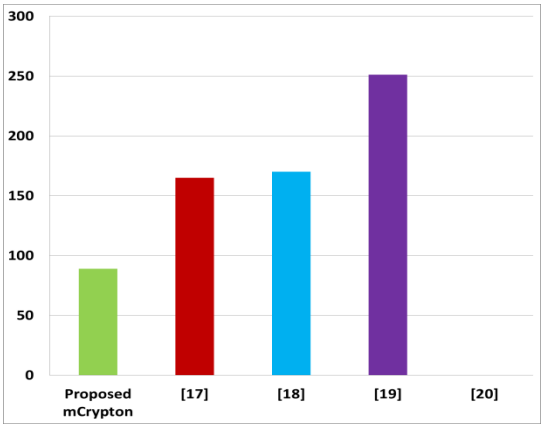

Figure 9. Power (mWatt) results

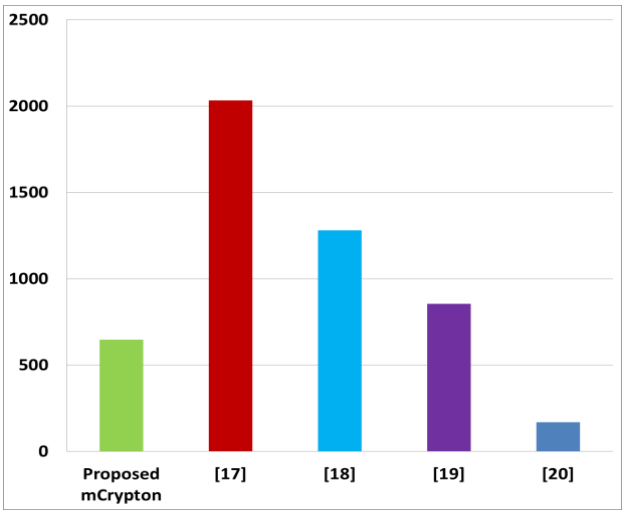

Figure 10. Throughput (Mbps) results

Table 1. Performance comparisons of mCrypton with previous studies

Algorithm	Block Size	Device	Max Freq. (MHz)	Thr/put (Mbps)	Total Slices	Efficiency (Mbps/Slice)	Power (mWatt)
Proposed mCrypton	64	Spartan-III	302	646	375	1.7	89
[17]	64	Virtex- 4FF668	31.765	2032	956	2.126	165
[18]	64	XC7Z010clq225-3	266.29	1280	521	2.45	170
[19]	64	Xilinx Zynq-7000	-	854	471	1.8	251
[20]	64	Xilinx Spartan-3	81.82	168.9	397	0.425	-

6. CONCLUSION

In this paper, efficient hardware architecture for the mCrypton lightweight encryption algorithm is introduced. The proposed architecture provides an optimization to the area and power consumption. The resource-shared structure has a good impact on area reduction. All components are designed to operate in a single clock cycle and a few numbers of slices. The implementation results using the Spartan-3Xilinx FPGA platform presented that only 375 slices are required to achieve 302 MHz of operating frequency with 89 mWatt power consumption. Further, a throughput of 646 Mbps and efficiency of 1.7 Mbps/slice is achieved. Thus, the obtained results proved that the proposed architecture is suitable for small and mobile devices.

REFERENCES

[1] S. Atiewi *et al*., "Scalable and Secure Big Data IoT System Based on Multifactor Authentication and Lightweight Cryptography," in *IEEE Access*, vol. 8, pp. 113498-113511, 2020, doi: 10.1109/ACCESS.2020.3002815.

[2] A. Shah and M. Engineer, "A Survey of Lightweight Cryptographic Algorithms for IoT-Based Applications," *Springer*, Singapore, 2019, vol. 669, pp. 283-293.

[3] D. Yang, W.-F. Qi, and H.-J. Chen, "Observations on the truncated differential of SP block ciphers and their applications to mCrypton and CRYPTON V1.0," *IET Information Security*, vol. 12, no. 5, pp. 419-424, 2018, doi: 10.1049/iet-ifs.2017.0196.

[4] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann and L. Uhsadel, "A Survey of Lightweight-Cryptography Implementations," in *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 522-533, Nov.-Dec. 2007, doi: 10.1109/MDT.2007.178.

[5] 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," in *IEEE Access*, vol. 9, pp. 28177-28193, 2021, doi: 10.1109/ACCESS.2021.3052867.

[6] H. Tong, J. Huang, and C. Qi, "A Novel Lightweight Cryptography Scheme Based on Standardized IOT Data," *Proceedings of the 2020 International Conference on Cyberspace Innovation of Advanced Technologies*, 2020, pp. 379-386, doi: 10.1145/3444370.3444601.

[7] L. Ning, Y. Ali, H. Ke, S. Nazir and Z. Huanli, "A Hybrid MCDM Approach of Selecting Lightweight Cryptographic Cipher Based on ISO and NIST Lightweight Cryptography Security Requirements for Internet of Health Things," in *IEEE Access*, vol. 8, pp. 220165-220187, 2020, doi: 10.1109/ACCESS.2020.3041327.

[8] F. Wu, L. Xu, X. Li, S. Kumari, M. Karuppiah and M. S. Obaidat, "A Lightweight and Provably Secure Key Agreement System for a Smart Grid With Elliptic Curve Cryptography," in *IEEE Systems Journal*, vol. 13, no. 3, pp. 2830-2838, Sept. 2019, doi: 10.1109/JSYST.2018.2876226.
[9] W. J. Buchanan, S. Li, and R. Asif, "Lightweight cryptography methods," *Journal of Cyber Security Technology*, vol. 1, no. 3-4, pp. 187-201, 2018, doi: 10.1080/23742917.2017.1384917.
[10] M. A. M. Isa, M. M. Ahmad, N. F. M. Sani, H. Hashim, and R. Mahmod, "Cryptographic key exchange protocol with message authentication codes (MAC) using finite state machine," *Procedia Computer Science*, vol. 42, no. 2014, pp. 263-270, 2014, doi: 10.1016/j.procs.2014.11.061.
[11] P. Yalla and J. Kaps, "Lightweight Cryptography for FPGAs," *2009 International Conference on Reconfigurable Computing and FPGAs*, 2009, pp. 225-230, doi: 10.1109/ReConFig.2009.54.
[12] A. K. Sahu, S. Sharma, and D. Puthal, "Lightweight Multi-party Authentication and Key-Agreement Protocol in IoT based e-Healthcare Service," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 2s, 2020, doi: 10.1145/3398039.
[13] A. Alamer, B. Soh, A. H. Alahmadi and D. E. Brumbaugh, "Prototype Device With Lightweight Protocol for Secure RFID Communication Without Reliable Connectivity," in *IEEE Access*, vol. 7, pp. 168337-168356, 2019, doi: 10.1109/ACCESS.2019.2954413.
[14] M. Sbeiti, M. Silbermann, A. Poschmann and C. Paar, "Design space exploration of present implementations for FPGAS," *2009 5th Southern Conference on Programmable Logic (SPL)*, 2009, pp. 141-145, doi: 10.1109/SPL.2009.4914893.
[15] F. Standaert, G. Piret, G. Rouvroy and J. Quisquater, "FPGA implementations of the ICEBERG block cipher," *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, 2005, pp. 556-561 Vol. 1, doi: 10.1109/ITCC.2005.155.
[16] P. Chodowiec and K. Gaj, "Very Compact FPGA Implementation of the AES Algorithm," in *International workshop on cryptographic hardware and embedded systems*, Springer, 2003, pp. 319-333, doi: 10.1007/978-3-540-45238-6_26.
[17] Y. A. Abbas, R. Jidin, N. Jamil, M. R. Z'aba, and M. E. Rusli, "PRINCE IP-Core on Field Programmable Gate Arrays (FPGA)," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 10, no. 8, pp. 914-922, 2015, doi: 10.19026/rjaset.10.2447.
[18] S. M. Soliman, B. Magdy and M. A. Abd El Ghany, "Efficient implementation of the AES algorithm for security applications," *2016 29th IEEE International System-on-Chip Conference (SOCC)*, 2016, pp. 206-210, doi: 10.1109/SOCC.2016.7905466.
[19] P. Ahir, M. M.- Kermani, and R. Azarderakhsh, "Lightweight architectures for reliable and fault detection Simon and Speck cryptographic algorithms on FPGA," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 4, pp. 1-17, 2017, doi: 10.1145/3055514.
[20] A. Mhaouch, W. Elhamzi and M. Atri, "Lightweight Hardware Architectures for the Piccolo Block Cipher in FPGA," *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2020, pp. 1-4, doi: 10.1109/ATSIP49331.2020.9231586.
[21] A. A. Alharith and N. R. Obeid. "Efficient Implementation for PRINCE Algorithm in FPGA Based on the BB84 Protocol," *Journal of Physics: Conference Series*, vol. 1818, no. 1, pp. 012216, 2021, doi: 10.1088/1742-6596/1818/1/012216.
[22] C. H. Lim and T. Korkishko, "mCrypton – A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors," in *Information Security Applications*, 2006, pp. 243-258, doi: 10.1007/11604938_19.
[23] M. Shakiba, M. Dakhilalian, and H. Mala, "Cryptanalysis of mCrypton-64," *International Journal of Communication Systems*, vol. 28, no. 8, pp. 1401-1418, 2015, doi: 10.1002/dac.1248.
[24] K. Jeong, *et al*., "Weakness of lightweight block ciphers mCrypton and LED against biclique cryptanalysis." *Peer-to-Peer Networking and Applications*, vol. 8, no. 4, pp. 716-732, 2015, doi: 10.1007/s12083-013-0208-4.
[25] H. Mala, M. Dakhilalian, and M. Shakiba, "Cryptanalysis of mCrypton-A lightweight block cipher for security of RFID tags and sensors," *International Journal of Communication Systems*, vol. 25, no. 4, pp. 415-426, 2011, doi: 10.1007/11604938_19.