The Performance of Various Lightweight Block Ciphers FPGA Architectures: A Review

Marwa Subhi Ibrahim[*], **Yasir Amer Abbas**[**], **Mudhafar Hussein Ali**[***]

[*, ***] *Computer Engineering Department, Engineering College, Al-Iraqia University, Iraq*
[*,**] *College of Engineering, Diyala University, Diyala, Baquba, Iraq*

Abstract

Today most of our devices connected with Internet to assistance us to improve our decisions. The number of people are using wireless and Internet networks increased day by day, which this increased improved the encryption mechanisms for devices and protect user data transfer over an unsecured network. Due to the limited resources for most portable devices, the concept of ubiquitous computing presents must be working terms of security, which contains Confidentiality, Integrity, Authentication, and non-repudiation. In comparison to energy-efficient with cryptography the conventional approaches are expensive and complicated and high-power consumption. The design of lightweight cryptography has solved big number of problem for hardware implementation with the conventional cryptography. In this paper, performance and efficiency depend of architectures review for lightweight block cipher algorithm base of FPGA design and implementation.

Keywords- FPGA, Lightweight, Block Cipher.

I. INTRODUCTION

In the past, there was no concern about encrypting for the communication devices such as actuators and sensors. Confidentiality of end-to-end communication is vital. Data read from smart meters can determine the occupancy of a home [1][2] since the information collected can be used by a thief to determine the best time to break into the home, for example. Most sensors and actuators still have limited hardware. Therefore, in recent years, several academic papers proposing lightweight cryptography focused for developed the lightweight encryption algorithms, lightweight encryption is encryption that has been tailored for implementations in constrained environments [3]. These restrictions can be characteristics such as the area of the chip, the consumption of the power, the size of the memory, or bandwidth for communication. It can also be seen that standardizing organizations are also interested in this type of encryption [4]. Authentication is also important to validate messages when commands are sent to actuators. Turning a lamp on or off may not be harmful, but turning off a life support system can have serious consequences. Communication protection must also work against message spoofing attacks [5]. A third important point in the IoT environment is the variety of devices – there are simple devices like a door sensor or more complex ones like a facial recognition camera. Some send information in the order of bits and others in the order of gigabytes. While a large block size is more efficient at transmitting a lot of data, it may be inappropriate for an element that sends little data and has a large hardware constraint [6]. The same goes for key size, for example to control home lighting, a 128-bit key is acceptable, and to control industrial components of a refinery, due to the risk of serious accidents in case of invasion, you can opt for larger keys. Therefore, the flexibility to work with variable block and key sizes is very interesting in this environment.

II. RELATED WORK

Internet-of-Things (IoT) era has brought a considerable increase in the number of interconnected devices, and consequently the volume of data shared between them. In addition to the greater number of devices, each year the computing power of these devices also increases, and their size and price decrease. The growth in the number of messages exchanged containing the most varied types of information generates a significant concern in relation to the security and privacy of users [7]. As a result, adequate information security techniques are being studied and developed for this new scenario. Proved that to ensure consistency, it is necessary to build an efficient ciphering algorithm that is both efficient and secure [8]. The concept of light weight cryptography was implemented and utilizing with variety FPGA boards. By comparing the algorithms to earlier work, it is possible to demonstrate that the modern algorithms for cipher block has increased throughput, makes more efficient use of hardware logic resources, and is more resistant to the majority of cryptanalysis assaults [9].

The Extended Tiny Encryption Algorithm (XTEA) implementing implemented to overcome the obstacles, at that time which was considered the fastest encryption algorithm with minimal operation code and the block cipher is a simple. The XTEA is a

http://doi.org/10.33193/IJSER.1.1.2022.43

Lightweight Block Cipher (LWBC) which has a round function operation of XTEA is in the form of a Feistel structure with 64 rounds for encryption and decryption which includes 64-bit block size (Plain text) and 128-bit key size. The key size is enough to adapt to the modern security requirements. The 128-bit Key is divided into four parts each is 32-bit wide; i.e., K0, K1, K2, and K3 and perform both encryption and decryption Process [10].

The Authors at [11] which presented the PRESENT algorithm that was symmetric encryption algorithm. The PRESENT algorithm is likewise called substitution permutation network (SPN) which has a block size of 64 bits and furthermore has two different key sizes: 80 or 128 bits. From here it was alluding to rendition with an 80-bit key as PRESENT-80 and the one with a 128-bit key as PRESENT-128. The PRESENT algorithm has 31 regular rounds and a final round that only consists of the key blundering step. One regular round consists of a key blundering step, a substitution layer, and a permutation layer.

Additionally, (Biswas et.al 2020) suggested a set of new criteria for evaluating block ciphers on an equal footing The LRBC has used the structural advantages of both substitution–permutation network (SPN) and feistel structure (FN) together to achieve better security. Hence, the along plaintext bit had been split into the several blocks each with 16-bit length of data which is being processed using a data-path and the same procedures will be repeated for consecutive data blocks. When all the data blocks are being processed, the results had been merged for produced the final encrypted text. The four number of keys with 4-bit each has been combined to each other and design 24 combination, the lightweight resource constrained block cipher strategy with a small and simple step had been designed. Generally, the feistel structure uses a large number of rounds and only operates on half of the block. SPN structure applies confusion, diffusion strategy which increases the redundancy of plaintext and confirms a strongly encrypted ciphertext. Thus, the mixture of this two structure has resulted in a more shielded system than using those strategies distinctively. The main constraints for designing lightweight encryption algorithm are storage, power, memory and speed [12].

LED is a symmetric block cipher whose block size is 64 bits and its internal architecture is based on the substitution permutation network (SPN). It is designed in two versions based on the key size; 64-bit key (LED-64) and 128-bit key (LED-128). Its number of rounds is based on the size of the encryption key; LED-64 has 32 rounds while LED-128 has 48 rounds [13].

GOST is Lightweight block cipher algorithm, which has a structure similar to DES algorithm, encrypts the 64-bits blocks with the 256-bits key. It has a fiestel network structure and data encrypted with an iterative way in 32 rounds. Since the computational time of encryption algorithms are very high, to make a real time and fast encryption algorithms Decryption done in 32 rounds and at the end of 32. round plain text is generated. GOST algorithms is developed alternatively for DES algorithm [14].

The mCrypton algorithm is a 64-bit lightweight block cipher cryptographic algorithm. It has Substitution permutation (SP) structure which 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 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. Radio Frequency Identification Device (RFID) tags, Internet of Things (IoT), and wireless sensor-networks (WSNs) are just a few examples of low-resource devices that might benefit from lightweight cryptography [15].

The XXTEA algorithm also known as Corrected block TEA was designed to improve the security characteristics of TEA family of ciphers. This algorithm is a feistel block cipher and has an unbalanced feistel network. It can work on variable length plain text and the minimum length of the block size should be 64 bits. The key size of XXTEA is of 128 bits. XXTEA uses simple addition, shift and XOR operations. The addition is modulo 32-bit addition operation. It uses a nonlinear function which improves the confusion and security characteristics of the cipher. The ability to work on variable length plain text makes it more efficient and practical to work on larger block size plain texts [16].

III. TYPES OF BLOCK CIPHER

In this part, we provide a brief summary of each block cipher that has been implemented from the design perspective (without explaining the key schedule) and the cryptanalytic perspective, where we restrict our state-of the art in the scenario of unknown key-settings and associated key settings [17].

1.1 AES

The Advanced Encryption Standard (AES) is a symmetric cryptographic technique in which both encryption and decryption operations utilize the same secret key. There are three variants of AES, and each one uses a different bit version for the secret key. AES may be divided into three categories depending on how many bits make up the secret key: AES-128, AES-192, or AES-256. Each version's encryption and decryption processes go through a different number of cycles. AES assisted in lowering the amount of hardware resources. Additionally, the circuit's overall delay. As a result, the greater operating frequency results from the smaller delay. The AES algorithm design is implemented in VHDL, then it is simulated and synthesized using Xilinx 14.2's Integrated

Environment synthesis software. Finally, the actual operation of the AES algorithm is tested using Spartan 6 FPGA hardware. It can be shown from the work done in this study that the AES Algorithm was developed and implemented with less FPGA hardware, reduced power consumption, and better cryptosystem throughput. Due to the Spartan-6 FPGA's implementation of the AES algorithm, portability, application compatibility, and implemented security levels are all possible [18].

1.2 KASUMI

KASUMI architectures are executed on the FPGA Xilinx Virtex 7, and they are recommended for wireless applications because to their high speed and compactness. The FI function's optimizations using CSB for S9/S7 and less combinational logic led to a relatively small KASUMI implementation. With crucial route reductions, a highly optimized pipeline layout easily increased throughput and efficiency for high-speed deployments. These layouts are ideal for networks that use wireless communication. In UMTS, GSM, and GPRS mobile communications systems, the block cipher KASUMI is used. The UEA1 and UIA1 confidentiality (f8) and integrity (f9) algorithms in UMTS employ the KASUMI algorithm [19]. This design's benefit is the KASUMI block cipher, which has not yet been cracked. Nowadays, the majority of identification systems need small-size hardware ciphers, therefore much effort has gone into and continues to go into developing an optimized hardware implementation for cryptographic ciphers like KASUMI. The drawback of KASUMI is that although a 128-bit key is accepted in the specification, there are certain circumstances when the key length must be shortened. The last two rounds of KASUMI application should be specifically created to protect against fault injection [20].

1.3 XXTEA

The Tiny Encryption Algorithm (TEA) is built with a basic, flexible hardware architecture that requires fewer calculations and easier key scheduling. An Extended TEA (XTEA), with pipelined architecture and parallel processing to increase throughput and give greater security, is created to combat security attacks in key scheduling on TEA. By altering the mode to conduct encryption or decryption, this XTEA algorithm is reconfigurable in nature. The TEA and XTEA simulation-results are implemented on the FPGA Platform-Artix-7 using the Xilinx ISE tool on the ModelSim (6.5f) simulator.

The XXTEA algorithm uses an array of 32-bit integers (at least two integers) and a 128-bit key to work, but it doesn't provide how to convert between bites and array. This causes multiple XXTEA implementations to be compatible with one another. Longs2bytes and bytes2longs handle the conversions between bites and array in this implementation. The input bytes are padded to multiples of 4 bytes (the size of a 32-bit integer) and are at least 8 bytes long, these precautions enable to encrypt every binary byte, regardless of length, as well as messages. This design's benefit is that it works well with wireless communication systems. Hardware methods for XXTEA implementation have drawbacks that may be defined for FPGA or reconfigurable devices. The highest encryption rate is provided by an FPGA system, but flexibility is sacrificed A fully new circuit layout is necessary for even minor circuit modifications [16].

1.4 RoadRunnerR

Roadrunner is a balanced Feistel-based network that supports 128/80-bit keys with a key length of 128 bits, 64-bit plaintext, and a total of 12 or 10 rounds, depending on the key size. The cipher is specifically designed to have a very small code size, the lowest possible latency, high throughput, and proven security, measured by the lowest possible number of active S-boxes in differential and linear trails. The RoadRunneR architecture has been implemented utilizing data-path sizes of 8-bit, 16-bit, and 32-bit. Various metrics, including power consumption, throughput, the number of GEs and LUTs needed, efficiency, clock cycles, and delay, are used to evaluate all of these data-path architectures. It is highly well-optimized for software implementation, tiny, quick, and has security against a variety of cryptographic attacks [21]. The comparison demonstrated the efficacy of various ciphers for an application and sometimes for scholarly reasons. A simple comparison of area or throughput figures is not sufficient nor fair since each platform and application has its own restrictions.

1.5 LED

The underlying architecture of the symmetric block cipher LED, which has a block size of 64 bits, is based on the substitution permutation network (SPN). Based on key size, it is created in two versions: (64-bit) key (LED-64) and (128-bit) key (LED-128). The amount of the encryption-key determines how many rounds it has; (LED-64) has 32-rounds, while (LED-128) has 48-rounds.

Iterative round-based mode or serialized nibble-wise mode may both be used to design the architecture-of the lightweight LED. The lightweight block cipher LED-64's round-based architecture. The design uses a (64-bit) input block and (64-bit) key size and is based on the LED-64. This architecture's internal functions are entirely built in Verilog HDL with both low-cost and high-end Altera and Xilinx FPGAs in mind. Both Xilinx ISE and Altera Quartus II with ModelSim for Altera FPGA-devices are utilized as the synthesis and simulation tools. Demonstrates the round-based (LED-64) architecture's block diagram. The benefit of this design is that it makes excellent candidates for lightweight-applications; for instance, these implementations provide the best area of all lightweight hash-function implementations that have been disclosed so far. RFID tags, WSN nodes, and smart cards are just a few examples of lightweight gadgets that are becoming more and more widespread in our everyday lives. These smart, lightweight gadgets might potentially modify critical information, necessitating security measures [13].

1.6 PRESENT

The most well-known lightweight block cipher shown at CHES 2007 is called "PRESENT." It encrypts blocks of 64 bits in length using keys of 80 or 128 bits. There are 31 rounds in all. A sub-key addition, an S-box layer that always calls the same nibble S-box, and a bit permutation layer make up the round function's basic SPN network. Due of highly particular linear biases, Security PRESENT has garnered a lot-of cryptanalytic interest. The studies examine PRESENT's linear behavior with relation to multiple linear trails. By employing the whole codebook, this kind of cryptanalysis enables the mounting of multi-linear-attacks on up to 27 rounds of PRESENT. Additionally, two different with complexity complexities about equivalent to those found from a thorough key search against the two versions of PRESENT are suggested in [11]. Implementation keep the key and the block to cipher in tables of 16-bit values. This design has the benefit of using the Present [22] method to maintain confidentiality in constrained environments.

1.7 PICCOLO

The Piccolo block cipher is a lightweight hardware-based block cipher. The memory and computing resources of hardware are constrained. The implementation for carrying out a number of trade-offs between area and speed. Using two alternative FPGA architectures the iterative and the 4-bit serial architectures. The Piccolo block cipher algorithm implemented with a 128-bit key. On the Xilinx Spartan-3, this algorithm's implementation was carried out. The resource usage rate for the iterative implementation is 76%. The encryption or decryption is completed in 31 clock cycles with this implementation. Consequently, it yields a throughput of 151.1 Mbps. The area of the serial implementation was optimized to reduce costs. 54 percent of the available resources are used, and it requires 496.

Table 1 provides a comparison of the findings on the hardware effectiveness of lightweight block ciphers with keys larger than 64 bits. In contrast to previous Feistel-type structure-based lightweight block ciphers, Piccolo offers both good security and incredibly compact implementation. Piccolo provides an adequate degree of security against known analyses, current related key differential attacks, and MITM attempts. The benefit of this design is that it was created with a minimum amount of hardware implementation and is appropriate for passive RFID tags. The Piccolo block-cipher's architectural investigation and serial and iterative implementation. Due to the fact that new applications need a lot of hardware resources and energy, PICCOLO's performance is not sufficient and trustworthy in this regard [23].

1.8 mCrypton

A 64-bit lightweight block cipher called mCrypton is developed for low-cost and resource-constrained applications like RFID tags and WSN sensors. According to the creators, mCrypton is resistant to both differential and linear cryptanalysis [4]. On eight rounds of mCrypton-128, however, a related-key rectangle attack has just been introduced in [5]. The attack has a success rate of 0.94 and requires roughly 246 plaintexts, 246 encryptions, and 5248 bytes of memory, respectively. To utilize the parameters of a 64-bit block length and variable key lengths of 64 bits, 96 bits, and 128 bits since the major benefit of building mCrypton is to provide a block cipher optimized for resource-constrained applications. A significant volume of bulk data encryption is either unneeded or even impossible. The drawback of building using Crypton's general architecture while redesigning and simplifying each component function to allow for considerably more compact implementation in both hardware and software [15].

1.9 ANU-II

The ANU-II algorithm is a 64-bit Feistel structure. That is the plaintext consists of 64 bit which is divided into two parts the most significant bit (MSB) of plaintext and the least significant bit (LSB) of plaintext. The ANU-II algorithm support two keys which are 128 bit and 80 bit. The cipher has 25 rounds. An effort has been made to get efficient design metrics for limited resources devices in IoT and RFID tags. The ANU-II cipher consists of the four Layers (S-Box, Circular right shift by 3, Circular left shift by 10 and XOR operation [24].

1.10 KLEIN

Sensors often have more hardware power and capabilities than RFID tags. Software-efficient block ciphers are seen to be more practical for sensors due to the flexibility and cost-effectiveness of software implementations in manufacturing and maintenance. KLEIN is a new family of block ciphers developed for devices with limited resources. KLEIN has the advantage of software performance on legacy sensor platforms over previous proposals, while at the same time, it’s hardware implementation may be small. According to the security study, KLEIN has a reasonable security margin against several cryptanalyses.

This design's advantage is that it can do three round functions: Sub Nibbles, Rotate Nibbles, and Mix Nibbles. This network is built on substitution-permutation and uses 64-bit plaintext and 64/80/96-bit variable keys to encrypt the plaintext in 12/16/20 rounds, respectively. The only non-linear phase of the KLEIN cipher that offers sufficient security to devices with limited resources is Sub Nibbles. The drawback of lightweight block ciphers, which were mostly created on old sensor systems. To withstand against weak key attacks, KLEIN uses the Feistel-based structure in key scheduling [25].

IV. DISCUSSION

Faster than the asymmetric algorithm is the symmetric algorithm if compared to the stream cipher, the block cipher is slower. When compared to software and hardware algorithms, hardware has more efficiency and better throughput especially with FPGA platforms[26][27]. Different performance metrics such as maximum frequency, throughput, area and efficiency are used to make a comparison of LWC algorithm implemented with FPGA existed studies. Several lightweight block cipher algorithms including AES, KASUMI, XXTEA192, RoadRunnerR, LED, XTEA, Piccolo, PRESEN, mCrypton, and KLEIN are study in this paper. The study show that the hardware implementation of KASUMI consumes far more efficiency and frequency than other algorithms and the throughput is better from other algorithms we studied. As a result of the investigation, it has been showed the hardware implementation of XXTEA192 far less area than other algorithm and the throughput is better from other algorithms [28][29]. When used iterative architecture the area increased so that the cost will be increased, due to the design of iterative architecture so its increase the number of CLK so that the frequency, efficiency and throughput will decrease. Due to the design of pipeline architecture the area is reduced so that the cost will be decreased, and the frequency, efficiency and throughput will increase. Table -1 shows that the pipeline architecture has the highest throughput, frequency, efficiency and lowest number of slice when compared with the other architecture. So it's very clear that the performance of algorithms depending on the architecture types[30].

Table 1: Comparison of LWC algorithm Implemented FPGA

Algorithm	Key (bit)	Block (bit)	Area Slice	Max Frequency (MHz)	Throughput (Mbps)	Efficiency Mbps/Slices	FPGA	Architecture
AES [18]	128	128	4089	495.32	6340	1.55	Virtex 7	Iterative
KASUMI [20]	128	64	468	644.33	5154.64	1.10	Virtex-5	Internal
XXTEA192 [16]	128	192	49	364.52	833.197	17.00	Spartan 6	Pipeline
RoadRunner [21]	128	64	404	105.80	17.711	0.043	Spartan-3	Iterative
LED [13]	64	64	122	485.79	971.51	7.96	Kintex-7	Iterative
XTEA [10]	128	64	238	263.76	80.43	0.34	Artix-7	Pipeline
Piccolo [23]	128	64	584	73.21	151.1	0.294	Spartan-3	Iterative
PRESENT [22]	80	64	152	364.56	171.19	1.12	Virtex 4	Pipeline
mCrypton [15]	64	64	375	302.00	646	1.70	Spartan-3	Iterative
KLEIN [25]	64	16	145	388.20	2070.39	-	Spartan	Iterative

V. CONCLUSION

FPGA are very suitable for the prototyping ultimately reduces cost for the algorithm design. The purpose of this research is to performed a comparison of Ten LWC algorithm implemented with FPGA existed studies; AES, KASUMI, XXTEA192, RoadRunnerR, LED, XTEA, Piccolo, PRESENT, mCrypton, and KLEIN in terms of frequency, throughput, area and efficiency against differential architectures. An analysis provided by this article is iterative architecture require more hardware resources. a highly optimized pipeline configuration with critical path reductions readily improved the throughput and efficiency. The storage, and transfer of sensitive or otherwise important data, guaranteeing the security for the resources and services is among the top responsibilities. As a direct result of this, the need for cryptographic components may expand. Although these devices have limited resources, the ongoing need for smaller sizes and cheaper manufacturing costs necessitates the adoption of safe algorithms that are often used in other areas, but with less weight. Other fields often make these decisions. The use of lightweight cryptography is necessary for embedded system security. The experts working on this area have been addressing the most recent advancements and the most effective lightweight block cipher implementations on FPGA. In order to achieve this objective, researches were performed on both software and hardware block cipher implementations. The results of a comparative study were provided in terms of architectures design is show how well each concept worked, how much they cost, and how secure they were.

REFERENCES

[1] A. K. Sahu, S. Sharma, and D. Puthal, "Lightweight Multi-party Authentication and Key Agreement Protocol in IoT-based E-Healthcare Service," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 17, no. 2s, pp. 1–20, 2021, doi: 10.1145/3398039.

[2] A. Alamer, B. Soh, A. H. Alahmadi, and D. E. Brumbaugh, "Prototype device with lightweight protocol for secure RFID communication without reliable connectivity," *IEEE Access*, vol. 7, pp. 168337–168356, 2019.

[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," *J. Netw. Comput. Appl.*, vol. 58, pp. 73–93, 2015, doi: 10.1016/j.jnca.2015.09.001.

[4] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, "A review of lightweight block ciphers," *J. Cryptogr. Eng.*, vol. 8, no. 2, pp. 141–184, 2018.

[5] Nayancy, S. Dutta, and S. Chakraborty, "A survey on implementation of lightweight block ciphers for resource constraints devices," *J. Discret. Math. Sci. Cryptogr.*, pp. 1–22, 2020.

[6] P. Panahi, C. Bayılmış, U. Çavuşoğlu, and S. Kaçar, “Performance evaluation of lightweight encryption algorithms for IoT-based applications,” *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 4015–4037, 2021.

[7] D. J. Rani and S. E. Roslin, “Light weight cryptographic algorithms for medical internet of things (IoT)-a review,” in *2016 Online international conference on green engineering and technologies (IC-GET)*, 2016, pp. 1–6.

[8] A. A. Yazdeen, S. R. M. Zeebaree, M. M. Sadeeq, S. F. Kak, O. M. Ahmed, and R. R. Zebari, “FPGA implementations for data encryption and decryption via concurrent and parallel computation: A review,” *Qubahan Acad. J.*, vol. 1, no. 2, pp. 8–16, 2021.

[9] P. Yalla and J.-P. Kaps, “Lightweight cryptography for FPGAs,” in *2009 international conference on reconfigurable computing and FPGAs*, 2009, pp. 225–230.

[10] R. Anusha and V. Veena Devi Shastrimath, “LCBC-XTEA: High Throughput Lightweight Cryptographic Block Cipher Model for Low-Cost RFID Systems,” in *Advances in Intelligent Systems and Computing*, 2019, vol. 986, pp. 185–196. doi: 10.1007/978-3-030-19813-8_20.

[11] M. Sbeiti, M. Silbermann, A. Poschmann, and C. Paar, “Design space exploration of present implementations for FPGAS,” in *2009 5th Southern Conference on Programmable Logic (SPL)*, 2009, pp. 141–145.

[12] A. Biswas, A. Majumdar, S. Nath, A. Dutta, and K. L. Baishnab, “LRBC: a lightweight block cipher design for resource constrained IoT devices,” *J. Ambient Intell. Humaniz. Comput.*, pp. 1–15, 2020.

[13] M. Al-Shatari, F. A. Hussin, A. A. Aziz, G. Witjaksono, M. S. Rohmad, and X. T. Tran, “An Efficient Implementation of LED Block Cipher on FPGA,” *2019 1st Int. Conf. Intell. Comput. Eng. Towar. Intell. Solut. Dev. Empower. our Soc. ICOICE 2019*, pp. 9–13, 2019, doi: 10.1109/ICOICE48418.2019.9035193.

[14] H. Aktaş, “Implementation of GOST 28147-89 encryption and decryption algorithm on FPGA,” 2018.

[15] R. Anusha and V. Veena Devi Shastrimath, “Y. A. Abbas, A. S. Hameed, S. H. Alwan, and M. A. Fadel, ‘Efficient hardware implementation for lightweight mCrypton algorithm using FPGA,’ vol. 23, no. 3, pp. 1674–1680, 2021, doi: 10.11591/ijeecs.v23.i3.pp1674-1680.,” in *Computer Science On-line Conference*, 2019, pp. 185–196.

[16] C. Kella, Z. Mishra, and B. Acharya, “A Compact & Low Power Architecture of XXTEA192 Lightweight block cipher,” in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021, pp. 972–976.

[17] A. A. M. Ragab, A. Madani, A. M. Wahdan, and G. M. I. Selim, “Design, analysis, and implementation of a new lightweight block cipher for protecting IoT smart devices,” *J. Ambient Intell. Humaniz. Comput.*, pp. 1–18, 2021.

[18] H. Zodpe and A. Sapkal, “An efficient AES implementation using FPGA with enhanced security features,” *J. King Saud Univ. - Eng. Sci.*, vol. 32, no. 2, pp. 115–122, 2020, doi: 10.1016/j.jksues.2018.07.002.

[19] N. Wu, Z. A. Ali, M. M. Shaikh, M. R. Yahya, and M. Aamir, “Compact and high speed architectures of KASUMI block cipher,” *Wirel. Pers. Commun.*, vol. 106, no. 4, pp. 1787–1800, 2019.

[20] M. Madani and C. Tanougast, “FPGA implementation of an enhanced chaotic-KASUMI block cipher,” *Microprocess. Microsyst.*, vol. 80, p. 103644, 2021.

[21] P. Pachange and G. Bansod, “A fast and efficient datapath designs of lightweight cipher RoadRunner on FPGA’s for resource constrained environments,” in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2019, pp. 65–72.

[22] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, “Lightweight hardware architectures for the present cipher in FPGA,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 64, no. 9, pp. 2544–2555, 2017.

[23] A. Mhaouch, W. Elhamzi, and M. Atri, “Lightweight Hardware Architectures for the Piccolo Block Cipher in FPGA,” *2020 Int. Conf. Adv. Technol. Signal Image Process. ATSIP 2020*, pp. 5–8, 2020, doi: 10.1109/ATSIP49331.2020.9231586.

[24] N. H. Yousif, Y. A. Abbas, and M. H. Ali, “Lightweight ANU-II block cipher on field programmable gate array,” *Int. J. Electr. Comput. Eng.*, vol. 12, no. 3, p. 2194, 2022.

[25] P. Singh, B. Acharya, and R. K. Chaurasiya, “High Throughput Architecture for KLEIN Block Cipher in FPGA,” *IEMECON 2019 - 9th Annu. Inf. Technol. Electromechanical Eng. Microelectron. Conf.*, pp. 64–69, 2019, doi: 10.1109/IEMECONX.2019.8877021.

[26] Q. Tang and F. Du, *Internet of Things Security: Principles and Practice*. Springer, 2021.

[27] S. S. Dhanda, B. Singh, and P. Jindal, “Lightweight cryptography: a solution to secure IoT,” *Wirel. Pers. Commun.*, vol. 112, no. 3, pp. 1947–1980, 2020.

[28] R. A. F. Lustro, A. M. Sison, J. T. Labiano, and R. P. Medina, “A lightweight block cipher implementation in the resource-constrained internet of things,” in *Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering, WCSE 2019, International Workshop on Computer Science and Engineering (WCSE)*, 2020, pp. 776–782.

[29] S. Q. A. Al-Rahman, A. Sagheer, and O. A Dawood, “A Hybrid Lightweight Cipher Algorithm,” *Int. J. Comput. Digit. Syst.*, 2021.

[30] K. Vipin and S. A. Fahmy, “FPGA dynamic and partial reconfiguration: A survey of architectures, methods, and applications,” *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–39, 2018.