International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Lorentz Chaotic System key generation with Low Area FPGA Implementation using Present Security Algorithm

Srikanth Parikibandla, Sreenivas Alluri

Abstract: ***Recently, the study of lightweight symmetric ciphers has gained more importance because of high requirement in the services for security in the CCNs (Constrained Computing Environments): Wireless Sensor Network (WSN), Internet of Things (IoT). A lightweight cipher is a cryptographic algorithm which is used for low resource device, minimal area optimization, low power design and attains sufficient security level. Size of the key is considered as major challenges in the cryptographic algorithms, because it increases the complexity of the cryptographic algorithm. To overcome this issue and improve the security, Lorentz Chaotic System (LCS) based PRESENT architecture is introduced in this research. The PRESENT lightweight block cipher is selected due to it is most general and famous lightweight algorithms. Hence, the random numbers were generated for a key purpose by using an LCS circuit. The streaming data will be encrypt and decrypt by using this algorithm. In this research, the modified lightweight block cipher algorithm is called as LCS- PRESENT architecture. Finally, the performance of LCS - PRESENT architecture was evaluated by FPGA hardware utilizations such as Lookup Table (LUT), flip flop, slices, and frequency. The security level of LCS- PRESENT architecture was analysed based on encrypted and decrypted results in XILINX tool. The LCS- PRESENT architecture utilizes the FPGA device to attain maximum accuracy and throughput, such as 30 of LUTs, 115 of flip flops and 47 of slices from available sources compared to existing cryptographic algorithms.***

Index Terms: ***Cryptography, Lightweight symmetric ciphers, Lorentz Chaotic Circuit, PRESENT, and Wireless Sensor Network.***

I. INTRODUCTION

Lightweight ciphers are significantly used for secure communication in resource-constrained devices [1–4]. Previously, software confidentially is rather seen from a security perspective, to prevent reverse architecture. Program is encrypted utilizing a regular cryptographic primitive and decryption is done utilizing hardware implementation of the decryption techniques in an assumed secure area [5].

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Srikanth Parikibandla*, Assistant and Associate professor. Research Scholar in ECE Department of GITAM deemed to be University, Visakhapatnam, India,
Dr. Sreenivas Alluri, Associate Professor in the Department of Electronics and Communication Engineering, GITAM Deemed to be University, Visakhapatnam, India,

The combination of the functionality is authentication, confidentially and integrity that are authenticated in ciphers of transformations in cryptographic [6–8]. Conventional cryptographic algorithms are established for the security of critical applications with various rounds for encrypt and decrypt of the information. Key is much important in any type of cryptographic algorithm, which is the important parameter of security level for any cryptographic algorithm. Key generation module of the cryptographic algorithm should be designed more careful in order to ensure the security of any system and it consumes several computational steps for strong key without correlation with further generated key value. The powerful algorithm is necessary to increase the encryption key randomness [9].

The ciphers are classified as asymmetric cipher and symmetric cipher. Asymmetric ciphers provide sufficient security features. The asymmetric ciphers are more demand and costly. It was classified into 2 as block and stream ciphers. Block ciphers implement the stream ciphers. The protocols will not designed by the stream ciphers [10, 11]. The parameters of the block cipher are block size, number of rounds and key size. Lightweight ciphers are used for efficient applications such as Wireless Sensor Networks (WSNs) [13], Internet of Things (IoT) [14], devices in cyber-physical systems [15]. This research implemented the lightweight block cipher based cryptographic algorithm due to its simple operation scheduling technique [12]. In this research work, a PRESENT lightweight cryptographic algorithm with minimum hardware utilization with an LCS based key generation scheme is proposed and implemented in FPGA platform. This research proposed a lightweight cipher LCS-PRESENT architecture to overcome the security issues in tremendously constrained environments. The major objective of the proposed LCS- PRESENT architecture is to provide high encryption and decryption quality by utilizing minimum FPGA device. Lightweight cipher LCS- PRESENT architecture is developed on a Xilinx FPGA tool, which provides low device utilization. The main contribution of the proposed method is follows.

- Model and optimize FPGA with the four rounds of the Implemented PRESENT algorithm and the cipher chosen
- Choose a cipher from lightweight block cipher for the implementation.

- The main aim of LCS based PRESENT architecture is it reduces the FPGA device utilization and power consumption, and considering the key generation mechanism in the design.

The overview of the paper is organized as follows. Section II briefly reviews the existing methods of PRESENT algorithm. Section III defines the LCS-PRESENT architecture for the experimental work. Section IV shows the experimental results and the comparisons of the architectures. Section V discusses about the Conclusion of the proposed method

II. LITERATURE SURVEY

Several researchers discussed on PRESENT cipher algorithm in cryptographic technologies for security improvement at the software level. A brief review of some important contributions of the existing PRESENT algorithms is presented in this section.

Lara-Nino, et al [16] proposed PRESENT lightweight hardware cipher architecture implemented on FPGA platform. In this study, two alternatives have been studied to generate the round key needed by algorithm. 16-bit data path architecture with 128–bit key schedule was implemented. Here, 16-bit data path architecture with 80 –bit key was developed where an area or security trade-off can be established. The results were consistent for both LUT-4 FPGAs. In the case of the LUT-6 platforms can be noted how implementations in Spartan-6 FPGA. This method uses the minimum of LUT elements that was in counts of low slice. The serial architecture creates the size of the implementation and to registers the measurements of performance and evaluation.

Thorat et al [17] introduce the Bit Permutation Instruction (BPI) with PRESENT cipher in S-box for the function of non-linearity that is new hybrid method of the lightweight encryption. The BPI is less than of log(n) in the instructions was perform by the n-bit permutation. The performance of the software is evaluated by the hybrid system, an advanced Reduced Instruction Set Computer (RISC) machine and the Intel Processor (IP) whereas Cadence tool was utilized to analyze the hardware performance. In this work, the required instruction count only depends on the number of formed section. The proposed method was required more memory; it increases the system complexity.

C. A. Lara-Nino, et al [18] proposed area costs and energy of the lightweight cryptographic algorithms for authentication encryption in WSNs. Two symmetric ciphers are PRESENT and Advanced Encryption Standard (AES), two hash functions are SPONCHANT and SHA were utilized for generic compositions. All architectures were developed and analysed in an FPGA based WSN. The experimental outcome shows empirically the advantages of employing lightweight algorithms over generic alternatives for reducing the impact in the lifetime of WNS. But, most of the time the sensor node is idle and it consumes more power without performing any task.

Hengameh et al [19] introduced a PRESENT cipher model, it incorporates both encryption & decryption process by utilizing 80 bit, 128 bit key for 64-bit input data security at the hardware level. This execution operation needs 64 bits permutation layer as well as 16 S box layers of 4 bits. Hence, to perform the key scheduling 2 different sized key was selected and it was designed as 80 bit and 128-bit keys. In this study, 80- bit, 128-bit key based PRESENT cipher model was evaluated in terms of flip flop, LUT, throughput, frequency. For this research two different sizes of the key are required, it may increase the total key size.

Resource efficient and high performance Very Large Scale Integrated (VLSI) architectures for PRESENT cipher have been proposed by J. G. Pandey et al [20]. In this research, proposed architecture of PRESENT was made on Xilinx Virtex -5 XC5VFX70t FPGA devices based on LUT-6 technology. In the Proposed architectures have a latency of the 33 clock cycle and 306.84 MHz of maximum clock frequency and 595.08 Mbps of throughput. The proposed PRESENT architectures utilize the FPGA slices for delivering data security under a resource-constrained environment. The FPGA performances of the proposed PRESENT architecture were analyzed using low configuration FPGA device.

To overcome the above-mentioned problems, this research is introduced LCS symmetric key based PRESENT algorithm for improving power consumption and security level.

III. PROPOSED METHODOLOGY

This section describes the LCS based PRESENT algorithm, which is symmetric ultra-lightweight block cipher for lightweight cryptography. So, the LCS-PRESENT algorithm is creating for the implementation of constrained environment. The PRESENT cipher employs three essential operations which is a structure that includes plaintext, modified at each round to produce confusion and diffusion over the input data. Fig.1 depicts a block of the LCS-PRESENT architecture.

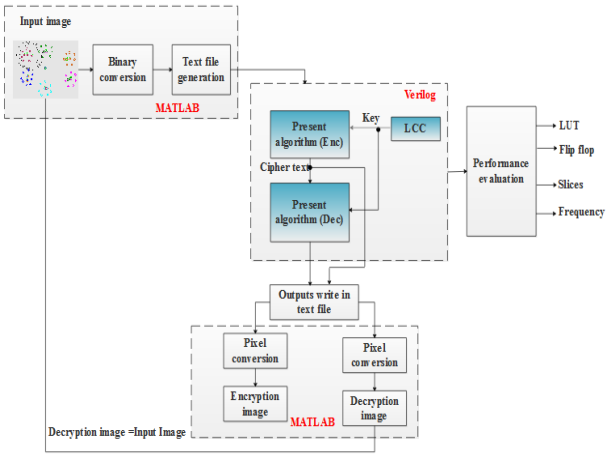

Fig.1 Block diagram of the LCS-PRESENT architecture

The principle for the LCS-PRESENT architecture is described as follows: Initially, an input image (128 ×128) is taken from Network Simulator-2 that denotes the deployment of nodes of the sensor in networks.

In the second step, the size of input image is 128× 128, so the total number of the pixel is 16384. Each pixel has 8-bit binary values. Fig. 2 shows an initial set of thirteen pixel's binary value.

In the third step, this binary value writes in a text file for giving the input to Verilog, because, the images are not possible to read in Verilog. So, pixel to binary conversion is performed in MATLAB. In this research, the proposed LCS - PRESENT architecture is designed in a Xilinx tool, which is described as follows

A. Proposed LCS- PRESENT Architecture

The proposed LCS - PRESENT design works on the block size of the 16 – bit. It supports 16 - bit key lengths. This LCS-PRESENT design depends on substitution and change system and it comprises of 4 rounds. Every one of the 4 rounds contains a XOR operation; it needs to present round key for $0 \le i \le 3$ in which is utilized for the post brightening task. This operation depends on straight bitwise stage layer and non-direct substitution layer. A solitary 4-bit S-box is utilized for the nonlinear layers, which is connected multiple times in parallel in each round for the most part four noteworthy capacities are required for the proposed LCS-PRESENT calculation, for example, key scheduling, AddRoundKey, S-box layer and, P-Layer. Key booking: it is a calculation which registers all round key from keys. AddRoundKey: Adds the state to a 16-bit word from the round key by using limited field math. S-box Layer: Creates 4 bit to bit substitution in the state utilizing S-box with 16-bit. P-Layer: Applies bit level moves over the state. Fig.2 demonstrates the information way for proposed LCS-PRESENT architecture.

11100101
11100101
11100101
11100101
11100101
11100101
11100101
11100101
11100101
11100101
11100101
11100101
11100101

Fig.2 Pixel format into binary conversion

The proposed PRESENT architecture (Fig. 3) consists of 16-bit encryption register 16-bit block is classified into 4-bit words (table 1). It is used to store the inward conditions of the encryption activity. A 64-bit state register is used to store the inward state alongside 16 – bit registers for putting away the halfway round key. Furthermore, 16-bit multipliers are used to switch the information between the heap stage and round calculation level. The information way contains a S box layer and one S-box for key booking. From this, one 16-bit XOR gate, 5-bit XOR entryway and 5-bit up counter are used. In this examination, the S-box is acknowledged by region improved Combinational Logic Circuit (CLC). The 16-bit register is used to get the figure content at the yield results. By this, the yield gets synchronized with the last round. The Latency can be diminished by one more clock cycle if this exploration does not require the yield to be registered. Though, the register is added to limit the control rationale, for synchronization of yield with last round. The registered output is obtained after completing 6 clock cycles by 4 rounds.

The major advantage of the proposed PRESENT architecture is a decrease in dormancy with proficient resource usage. In the initial clock cycle, load the plaintext, in the next clock cycle the multiplexer switches the input information and for 4 cycles every middle state are determined. The information is obtaining at the encryption register and mix with the middle round key. Moreover, the blended state is affirmed to the S-box layer for state preparing, it conveys 16-bit information parallel to P-Layer. It is passed to encryption register through the multiplier state for state provides, which gives 16-bit information. The input binary values have been stored in the LUT- box in the LCS-PRESENT architecture. For example, 0 to 15-bit binary values are stored in hexadecimal form in the S1 [A]. The LUT- box has the 4-bit binary value of 2 that provides the output 4-bit (0110) with the 4-bits S-box S1 in hexadecimal form is shown in the tab. 1 and PRESENT encryption P_ box bit. The 16 - bit binary values are stored in hexadecimal form in P [A] which shown fig 2.

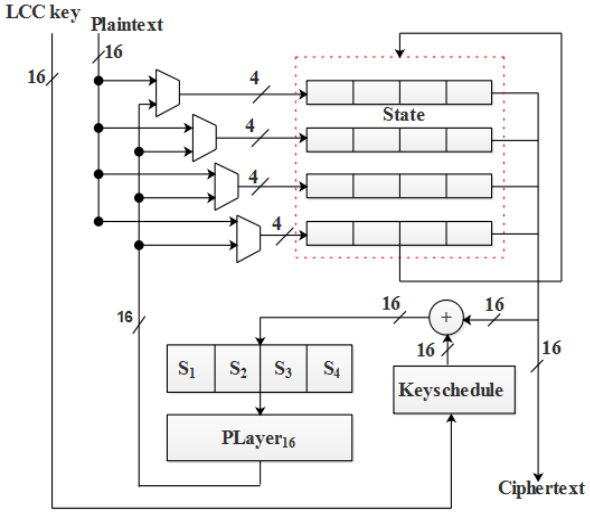

Fig.3 Data path for proposed PRESENT architecture

The output register is present in the cipher at the last clock cycles. The LCS-PRESENT cipher model involves the description to acquire the input information. The decryption process is utilized for key scheduling. After performing the present Encryption and decryption process, the outputs are stored in a text file. These binary text files are read in the MATLAB tool to show the encrypted & decrypted image.

In each round, the 16-bit word key (table 2) is created from the furthest right four bits in the key registers, for this situation just 16-bit out of the 4-bit contained in the registers are utilized. Fig. 4 demonstrates the key scheduling for the proposed PRESENT architecture.

The proposed PRESENT architecture can process the 4-bit expressions of the state in 4-cycles. The additional change over the four 4-bit full registers is executed in the fourth cycle. This system gives an inactivity of four cycles for every round. Since four cycles are also required to produce cipher text output, the proposed architecture will take 4 cycles to process a 16 -bit plaintext block, depending on the nature of the key generation process. The architecture of key generation is briefly described as follows.

Table 1. Present encryption S_ box Four- bit S-box S1

A[i/p]	0	1	2	3	4	5	6	7
S1[A]	0	4	8	C	1	5	9	13
A [i/p]	8	9	A	B	C	D	E	F
S1[A]	2	6	A	8	4	7	1	2

Table 2. Present encryption P_ box 16- bit

A[i/p]	0	1	2	3	4	5	6	7
P [A]	C	5	6	B	9	0	A	D
A [i/p]	8	9	A	B	C	D	E	F
P [A]	3	E	F	E	2	7	B	F

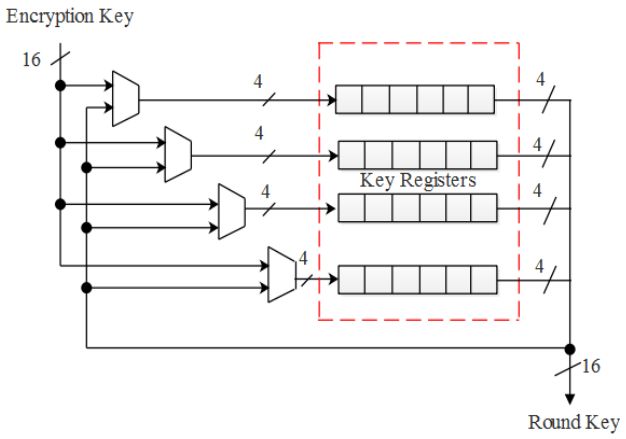

Fig.4 Key schedule for proposed PRESENT using 16 –bit

B. Architecture for Key Generation

In this research, key generation is the difference between the proposed PRESENT architectures and existing architecture. For the generation of round-keys, this research work explains an alternative way to process 16-bit input keys, which uses logic standard to provide each round key. The LCS used for generating the chaotic software key of the encryption & decryption process, which is shown in Fig. 5. The key management is a challenging task in a cryptographic, the key size is essential in area cryptosystem. A key size ensures the randomness, but it proportionally increases the load with complexity. To overcome this problem, the LCS based key used for PRESENT architecture. The random number is generated for software key in the LCS

First Stage

In initial stage, first stage output and second stage output is perform the addition process which is mentioned in eq.1. Then, some of the arithmetic operation has been performed and register 1 output is stored in X[n] which is mentioned in eq.2

$$R_1[n-4] = X[n] + Y[n] \quad (1)$$

$$Cmul_1[n-3] = R_1[n-4] \times M_1$$

$$Cmul_2[n-2] = Cmul_1[n-3] \times M_2$$

$$R_2[n-1] = Cmul_2[n-2] + X[n]$$

$$\text{Re}\,gister_i = R_i = R_2[n-1]$$

$$X[n] = \text{Re}\,gister_i \quad (2)$$

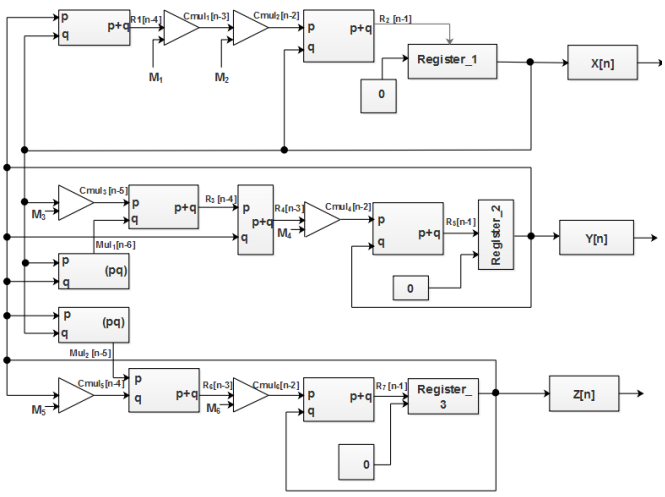

Fig.5 Block diagram of the Lorenz Chaotic System

Second Stage

The same 2 stage registers are performed the multiplication which mentioned in Eq.3. Register 2 value is stored in eq.4

$$Mul_1[n-6] = X[n] \times Y[n] \quad (3)$$
$$Cmul_3[n-5] = X[n] \times M_3$$
$$R_3[n-4] = Mul_1[n-6] + Cmul_3[n-5]$$
$$R_4[n-3] = R_3[n-4] + Y[n]$$
$$Cmul_4[n-2] = R_4[n-3] + M_4$$
$$R_5[n-1] = Cmul_4[n-2] + Y[n]$$
$$\text{Re}\,gister_2 = R_5[n-1]Y[n] = \text{Re}\,gister_2 \quad (4)$$

Final Stage

For the final stage, rest of the two stage connection also required to get the 3rd stage output which is given in Eq.6. The normal multiplication of two register output is mentioned in Eq.5

$$Mul_2[n-5] = X[n] \times Y[n] \quad (5)$$
$$Cmul_5[n-4] = Z[n] \times M_5$$
$$R_6[n-3] = Mul_2[n-5] + Cmul_5[n-4]$$
$$Cmul_6[n-2] = R_6[n-3] \times M_6$$
$$R_7[n-1] = Cmul_6[n-2] + Z[n]$$
$$\text{Re}\,gister_3 = R_7[n-1]Z[n] = \text{Re}\,gister_3 \quad (6)$$

Fig. 5 shows three different stages, which depends on one

stage to another. Each stage has two inputs such as ***p*** and ***q*** which are stored in various registers like Register1, Registe2 and Register3. The LCS outputs represent as X[n], y[n] and Z[n], these three outputs are connected to the MUX. The counter has been designed which output is the selection line for the MUX. If the counter output is 0, X[n] will be output. If counter is 1 and 2, Y [n] and Z [n] will be the output. So, each and every clock cycle the key value will be varied based on the 3 stage output. Some of the key generation also available in existing methods, but none of the design have this kind of three stages and will generate the randomized key like LCS. Due to usage of LCS, the random key is too difficult to identify from the unknown person. LCS has been successfully generated the random number compared to other chaotic circuits which is possible in digital designs. This architecture of key schedule performs by recording all the key material in model permitting the synthesis tool to generate the combination circuit design, which provides the round keys, it is much interesting for this exact design since size of round key is minimized from 128 to 16 bits, it enables a reduction in the complexity of PRESENT architecture.

V1.EXPERIMENTAL RESULT AND DISCUSSION

In this scenario, the proposed LCS-PRESENT architecture is implemented in Xilinx FPGA by using Verilog code language and synthesized using Xilinx Register Transfer Level (RTL) compiler for Virtex -6 FPGA devices on Xilinx platform. Device utilization of the LCS-PRESENT architecture is analyzed in Virtex-6 due to it is high configuration device. In this research, the verification of the proposed PRESENT designed system is verified by utilizing a design supporting tool like Xilinx tool and it simulated by utilizing Modelsim tool. The execution system gives significant results respective to outputs. The device utilization of the proposed PRESENT architecture is given in Table 3. In this research, the 16-bit key based PRESENT architecture cipher model is analysis the performance of encryption and decryption process by means LUTs, flip flips, slices, and frequency.

The LCS- PRESENT architecture is implemented in a FPGA platform. This platform is much suitable for VLSI implementations because of its low power, flexibility, and upward compatibility compared to the ASIC platform. Generally, the VLSI circuits for the bitwise algorithms require high performance and low latency under limited chip area and complexity. The circuits are commonly required to support the high data rates for the communication networks. Here, the proposed PRESENT architecture averagely utilizes 1 % LUTs from available source of 46,560, 1% of the flip flop from the available source of 46, 560 and 1% of slices from available source of 11, 640. In this research, PRESENT architectures 4 clock cycles in the latency the runs the 90.26 MHz frequency. The LCS- PRESENT architecture consumes 1.293 watts of power, which is shown in Fig.6. Along with the efficient utilization of the device resources and the performance of the architecture has also improved. This result proves that the proposed PRESENT architecture was much suitable for encrypt and decrypt the information process.

Table 3. Device utilization summary for Xilinx Virtex- 6 XC6VCX75t FPGA device for LCS-PRESENT architecture

Elements	Available Resources	Used in count	Utilization in percentage
Number of slice registers	93,120	131	1%
Filip Flop	46,560	115	1 %
Number of slice LUTs	46,560	130	1 %
Number of used as a logic	46,560	125	1 %
Slice	11,640	47	1%

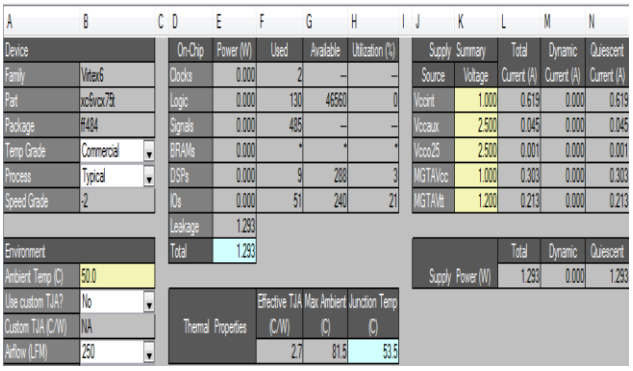

Fig.6 Screenshots of the power consumption by LCS-PRESENT architecture

Table 4 shows the comparison resource utilization for existing PRESENT and proposed LCS- PRESENT architecture on FPGA devices. The existing PRESENT architectures are synthesized for 2 various Xilinx devices namely Virtex-5 XC5vlx 50t 3ff 1136 [11], Virtex-5 XC5VLX50 [15] and Virtex -6 XC6LX 16-CS324 [13]. The proposed PRESENT architecture occupies 31.57 % FPGA LUTs, 24.83 % of flip flops and 29.85 % of slices compared to Lara-Nino et al. [11]. The proposed PRESENT architecture occupies 92.32 % of FPGA LUTs, 86.99 % of flip flops and 90.89 % of slices compared to Lara-Nino et al. [13]. The proposed PRESENT architecture requires 51.127 % lower FPGA LUTs, and 31.88 % lower slices compared to Lara-Nino et al. [15]. The synthesis results of the implementations are shown in Table 4. From the table, the proposed architecture gets averagely 90% to 92% lesser FPGA device utilization in comparison to the architecture of [11], [13] and [15]. Along with the efficient device utilization of resources, the performance of the proposed LCS-PRESENT architecture has improved because the size of the key is efficiently reduced. The proposed architecture has the ability to perform on frequency by 90.26 MHz compared to existing architectures [11], [13] and [15].

Table 4.Comparison resource utilization for existing PRESENT and LCS- PRESENT architecture on FPGA devices

Authors	Year	FPGA Devices	LUTs	Flip flops	Slices	Frequency (MHz)
Lara-Nino et al. [11]	2017	Virtex-5 XC5vlx 50t 3ff 1136	239	201	73	431.78
			190	153	67	543.30
Lara-Nino et al [13]	2018	Virtex -6 XC6LX16-CS324	1694	884	516	13.56
Pandey et al. [15]	2017	Virtex -5 XC5VLX50	266	-	69	306.84
Proposed LCS-PRESENT	2019	Virtex-6 XC6VCX75t	130	115	47	90.26

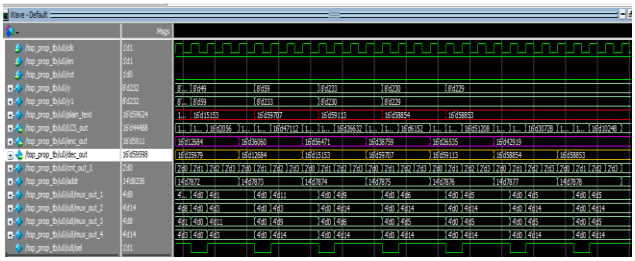

Figure.7 Output waveform of the LCS- PRESENT architecture

Fig. 7 represents the output waves of the LCS- PRESENT architecture. It is taken from modelsim. In Fig.7, red color represents input plaintext, light green represents LCS key, violet color represents enc_out (encryption output), yellow color represents dec_ out (decryption output).

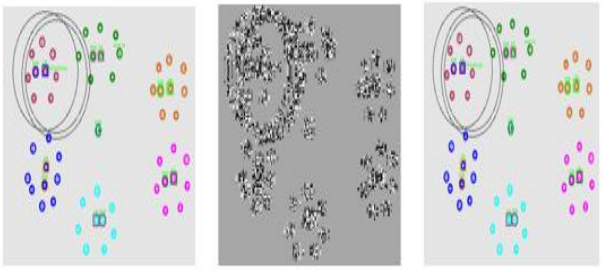

Figure. 8 Sensor node input image, encrypted image and decrypted image

Fig. 8 represents the image of sensor node sample input images (sensor node transmission image and image of node deployment), images of encryption and decryption. Here, the LCS- PRESENT architecture is tested using the two sensor node images. Fig. 8 shows the decrypted image which is obtained by the proposed architecture and obtained image is similar to the input image. From this, it clearly shows the input image is not affected in encryption process. The proposed LCS- PRESENT architecture provides high encryption and decryption security with minimum FPGA device utilization and power consumption.

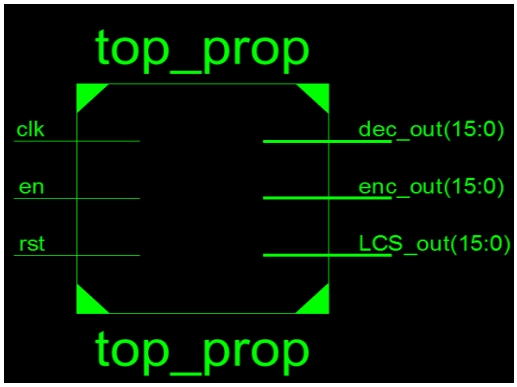

Figure.9 Top module of the LCS- PRESENT architecture

Fig. 9 depicts the Register Transfer Level (RTL) view of the top module for LCS- PRESENT architecture taken from the Xilinx software tool by using Verilog. In this research, the LCS- PRESENT architecture has an individual code for each block such as encryption, decryption, and LCS key schedule. Size of input image is 128×128 and each pixel is converted into binary. In this, each pixel size represents 8-bits and the entire depth of the image is 16384 bits. Fig. 9 consists of the clock signal (Clk), enable signal (ebl), reset signal (rst), dec_out (15:0), enc_out (15:0) and LCS_out (15: 0). Fig. 10 shows the internal block of the top module for the LCS-PRESENT architecture. In Fig.10, all the internal blocks are connected by using red color wire as the main module.

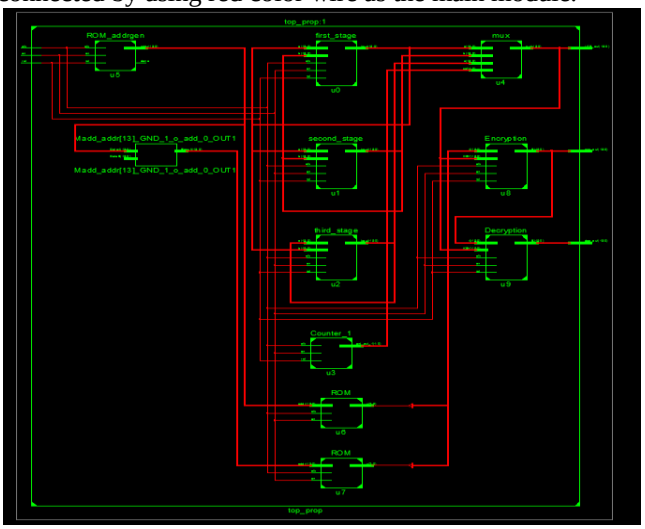

Figure.10 The internal module of the LCS- PRESENT architecture

V.CONCLUSION

This research proposed two significant Very Large Scale Integrated (VLSI) architectures for PRESENT cipher with a key size of 16-bit. The proposed PRESENT architecture efficiently utilizes the FPGA device (LUTs, flip flops, and slices) for providing data security under the asset-constrained condition. The LCS has used to generate the random key which is applicable for encryption and decryption. Because of the LCS key only, the data can be retrieved in decryption side. If there will be any chances in key, the original data won't be retrieved.

The proposed architectures efficiently utilize the FPGA slices for providing data security under the asset-constrained condition. The proposed design has been modeled in Verilog language and the architectures have been synthesized in the Xilinx Virtex-6 XC6VCX75t FPGA device. The presented architectures consume only 1 % of LUTs, 1 % of the flip flop and 1 % of slices form given resource utilization respectively. The proposed architecture proves an improvement in terms of less device utilization and less power consumption compared to other existing implementations, so it is suitable for utilizing in lightweight cryptography applications. In future work, different RTL will be used to generate the random number as well as efficient encryption VLSI architecture will be designed for further reducing the FPGA device utilization in the cryptography system.

REFERENCES

1. B. J. Mohd, T. Hayajneh, K. M. Ahmad Yousef, Z. A. Khalaf, and Md. Z. AlamBhuiyan, "Hardware design and modeling of lightweight block ciphers for secure communications", *Future Generation Computer Systems*, Vol. 83, pp. 510-521, 2018.
2. P. Bhoyar, S. B. Dhok, and R. B. Deshmukh, "Hardware implementation of secure and lightweight Simeck32/64 cipher for IEEE 802.15. 4 transceiver", *AEU-International Journal of Electronics and Communications*, Vol. 90, pp. 147-154, 2018.
3. R. Sadhukhan, S. Patranabis, A. Ghoshal, D. Mukhopadhyay, V. Saraswat, and S. Ghosh, "An evaluation of lightweight block ciphers for resource-constrained applications: Area, performance, and security", *Journal of Hardware and Systems Security*, Vol. 1, No. 3, pp. 203-218, 2017.
4. N. Thangamani, and M. Murugappan, "A Lightweight Cryptography Technique with Random Pattern Generation", *Wireless Personal Communications*, Vol. 104, No. 4, pp. 1409-1432, 2019.
5. T. Hiscock, O. Savry, and L. Goubin, "Lightweight instruction-level encryption for embedded processors using stream ciphers", *Microprocessors and Microsystems*, Vol. 64, pp. 43-52, 2019.
6. W. Diehl, and K. Gaj, "RTL implementations and FPGA benchmarking of selected CAESAR Round Two authenticated ciphers", *Microprocessors and Microsystems*, Vol. 52, pp. 202-218, 2017.
7. D. Chakraborty, and P. Sarkar, "On modes of operations of a block cipher for authentication and authenticated encryption", *Cryptography and Communications*, Vol. 8, No. 4, pp. 455-511, 2016.
8. F. Zhang, Z. Liang, B. Yang, X. Zhao, S. Guo, and K. Ren, "Survey of design and security evaluation of authenticated encryption algorithms in the CAESAR competition", *Frontiers of Information Technology & Electronic Engineering*, Vol. 19, No. 12, pp. 1475-1499, 2018.
9. C. Baskar, C. Balasubramaniyan, and D. Manivannan, "Establishment of light weight cryptography for resource constraint environment using FPGA", *Procedia Computer Science*, Vol. 78, pp. 165-171, 2016.
10. X. Fan, K. Mandal, and G. Gong, "Wg-8: A lightweight stream cipher for resource-constrained smart devices", *In International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pp. 617-632, 2013.
11. Aldabbagh, Sufyan Salim Mahmood, and ImadFakhriTaha Al Shaikhli. "OLBCA: A New Lightweight Block Cipher Algorithm." In 2014 3rd International Conference on Advanced Computer Science Applications and Technologies, pp. 15-20. IEEE, 2014.
12. C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Ultra-lightweight implementations for smart devices–security for 1000 gate equivalents", *In International Conference on Smart Card Research and Advanced Applications*, Springer, pp. 89-103, 2008.
13. L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey", *Computer networks*, Vol. 54, No. 15, pp. 2787-2805, 2010.
14. 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, 2015.
15. 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)*, pp. 1-6, 2013.
16. 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, 2017.
17. C. G. Thorat, and V. S. Inamdar, "Implementation of new hybrid lightweight cryptosystem", *Applied Computing and Informatics*, 2018.
18. C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Energy and Area Costs of Lightweight Cryptographic Algorithms for Authenticated Encryption in WSN", *Security and Communication Networks*, 2018.
19. D. A. Hengameh, P. V. Joshi, "An Efficient Implementation Of Present Cipher Model With 80 Bit And 128 Bit Key Over Fpga Based Hardware Architecture", *International Journal of Pure and Applied Mathematics*, Vol. 119, No. 14, pp. 1825-1832, 2018.
20. J. G. Pandey, T. Goel, and A. Karmakar, "An efficient VLSI architecture for PRESENT block cipher and its FPGA implementation", *In International Symposium on VLSI Design and Test*, pp. 270-278, Springer, 2017.

AUTHORS PROFILE

Srikanth Parikibandla received the B.Tech (EIE), M.Tech-ECE (VLSI Design) degrees in 2002 and 2007 respectively in regular mode. He is having 15 Years of teaching Experience in various Engineering Colleges as Assistant and Associate professor. He is currently Research Scholar in ECE Department of GITAM deemed to be University with special interests on Hardware implementation and improving the performance in data security, cryptography, and IoT devices. His research interests focus on the development of hardware/software security schemes for networked embedded systems and for the IoT Wireless Sensor Nodes.

Dr. Sreenivas Alluri received the B.E. degree in 2000, M.E. degree in 2004 and Ph.D degree in 2013, all in Electronics and Communication Engineering from Andhra University, Visakhapatnam. Since 2008 he has been an Associate Professor in the Department of Electronics and Communication Engineering, GITAM Deemed to be University, Visakhapatnam, India, His research interests include satellite Image, Video and Audio analysis and reconstruction, and the processing algorithms for digital signal, and image processing. At GITAM Deemed to be University he has been involved in the development Embedded system for Unmanned Vehicles.