An Ultra-High Throughput and Efficient Implementation of Advanced Encryption Standard

Sarita Sanap[1] and Vijayshree More[2]
[1] Maharashtra Institute of Technology, Dr. B.A.M. University, Aurangabad, India
[2] Jawaharlal Nehru Engineering College, MGM University, Aurangabad, India
Email: saritawagh1@gmail.com; vijayshreemore@gmail.com

***Abstract*—Encryption techniques have become most important in the digital world. Advanced Encryption Standard implementation enhances security. In this proposed work, advanced encryption standard implementation is done on field programmable gate array through minimum resource utilization. Experimental results obtained gives area-efficient, high-throughput hardware structure. To achieve a high throughput rate and minimize resource allocation, parallel-pipeline design and data forwarding mechanism with optimized S-box is proposed. Comparison between proposed work and existing work shows that this optimized implementation gives reduction in resources and increase in throughput. Proposed method achieves throughput of 97.11Gbps and efficiency of 85.18 Mbps/slice.**

***Index Terms*—Field programmable gate array, avalanche effect, throughput**

I. Introduction

With the advancement of technology, data protection have become one of the most important concern for communication systems in order to avoid deception**.** Preventing confidential information manipulation and unauthorized access to data is a high priority. Encryption technique is essential in data security because it protects against known threats and reduces the chance of information theft. The Advanced Encryption Standard is most extensively used symmetric key encryption techniques for securing data [1].

AES algorithm is used in wide range of applications which includes light fidelity network environment [2] smart grids [3] communication [4], defense [5], networking, social medias, E-banking, cloud computing [6], [7], healthcare [8], internet of things systems [9], [10]. Advanced encryption standard implementations can be divided into two categories as software and hardware. Hardware implementation of AES gives more physical security and speed. Software implementation achieves less throughput than hardware implementation. So it leads to high power requirement. So it is not suitable for constrained nodes. Field Programmable Gate Arrays (FPGAs) are suited for cryptographic algorithm implementation. They are reconfigurable and provide both time and cost effective solutions. Furthermore, field programmable gate arrays offer significantly improved speed capability. As modular arithmetic is done more efficiently in FPGAs, they are advantageous for implementing cryptanalytic attacks [11]. Application specific integrated circuits has a long development cycle, and significant computational expenses whereas FPGAs provides high security, high speed and flexibility [12]. Advanced Encryption Standard is a symmetric block cypher with a 128-bit data block and 128, 192, and 256 bit variable key sizes. AES is an iterative technique that works using a symmetric square matrix, commonly known as a state. The message is organized by the state. AES consists of main four modules as substitution bytes, shift rows, mix column and add round key [13]. In substitution module each entry in the state array is replaced with an element from the s-box by the substitution operation. The s-box is usually made of $GF(2^8)$, which is meant to protect against all threats. The S-box is invertible and is obtained by performing two transformations: first, a multiplicative inverse in $GF(2^8)$ with element 00 mapped to itself, and then an affine transform over $GF(2^8)$. It can also be implemented as an S-box, which is a look-up table with pre-computed values. Many researchers are focusing on design of efficient s-box as cipher text obtained is mainly dependent on S-box. Shift row transformation implies cyclic shift which helps in creation diffusion. Mix column process uses polynomials over $GF(2^8)$. It is multiplied by modulo and fixed polynomial. Add round key module gives round key on which XOR operation with the state is done. As Advanced Encryption Standard algorithm is used extensively, it is necessary to perform optimization process on it. If optimization is not done it makes the process more time consuming and hence increases the network use, power consumption, and delay in the network. Existing security systems employ one or two attributes at a time for optimization, hence low security and more time consumption to encrypt the data is observed. The algorithm therefore should incorporate

Manuscript received June 15, 2022; revised August 18, 2022; accepted September 18, 2022.
Corresponding author: Sarita Sanap (email: saritawagh1@ gmail.com).

©2022 Int. J. Elec. & Elecn. Eng. & Telcomm.

more attributes for optimization process. Hence proposed efficient algorithm aims at optimization of round and substitute byte both, which helps to achieve high throughput and less time consumption. The main contribution of this paper is to provide efficient implementations of AES algorithm on FPGA. Proposed method aims at designing of optimized AES modules by considering optimal use of resources. By using pipelining method and optimized modules implementation of proposed method gives significant improvement in throughput and efficiency.

The remaining paper is organized as follows. Section II summarizes literature study related to implementation of AES. Section III presents the optimization and resource mapping approaches for the AES algorithm implementation. Section IV discusses the experimental results and comparative analysis based on obtained parameters. Section V of the paper concludes with a summary of the work.

II. Related Work

As AES is suitable for wide range of applications, exploration of various architecture based on different modules is done. Recently multiple FPGA implementation have been proposed. Author had proposed new positive polarity reed Muller (MPPRM) architecture as using combinational logic circuitry for implementation substitute bytes and Inverse Substitute bytes transformation [14]. It reduces hardware requirement. FPGA devices Kintex 7, Spartan 6, Virtex 5, and Virtex 6 are used for implementation by the author. However key expansion architecture which minimize the time consumption during key generation is not proposed in this work. Modification in AES process is done by author by combining add-round key and shift-rows a, for the first nine rounds. Shift-rows function is not added in last round. Xilinx XC5VLX85-FF676-3 device is used for implementation. But lower throughput is reported in this work. [15]. Author proposed composite field arithmetic based S-Box operation. Pipelined S-box which is MUX based gives balance between area and throughput. The structure is implemented on the FPGA Vitex-4 device. [16]. For IoT application lightweight AES is required so optimization of S-Box is done by the author. Also inside state register shift-row operation is embedded. To reduce power consumption clock gating in different blocks is used. Virtex 5 device is used [17]. Modification in the subkey generation architecture is done by the author to increase the speed of key generation process. The proposed architecture is simulated and is implemented in FPGA virtex 5 XC5VLX50T. FPGA is used because it is a reconfigurable device and it gives better results [18]. Reduction of finite field GF(2^8) into GF(2^4) is proposed by author. However further time consumption can be reduced by implementing the pipelined architecture by the author. This process of reduction the finite field helps in simplification of both the software and hardware design. Improvement in speed up the process of generating and looking up the entries of the S-box and inverse S-box is obtained [10]. Author proposed a key generation block based on the PN Sequence generator to generate the initial key. The key does not need to be applied externally. But pipeline method is not proposed by author. The implementation of AES algorithm with modified S-box values using Spartan6 XC6SLX150-3FGG900 FPGA [19]. For improvement of security and reliability author had proposed AES round architecture into three parts and two pipelines registers are added in between. Fault injection simulations are done here. Implementation done on Virtex 5 using Xilinx integrated synthesis environment 13.4. However compared to the conventional AES, without fault detection system architecture, the throughput is lower [20]. Author had focus on fast processing of AES for IoT devices. In this, the AES algorithm is implemented on Spartan-6 and Virtex 6. Spartan 6 provides better throughput and less time delay which is helpful for IoT devices. Lower throughput of 1.44 Gbps and 1.99 Gbps for Virtex-6 and Spartan-6 respectively is reported in this work [21]. Author had used efficient multipliers in GF(2^8). Full pipelining technique is proposed. However high power consumption is reported. High-throughput AES synthesis in ECB mode using Virtex-5 (XC5VLX85-FF676-3) and Virtex-6 (XC6VLX240T-FF784-3) is done [1]. Many researcher contributes for S-box for optimization of AES. Composite field arithmetic is used for S-box implementation, but proposed mix column transformation using vedic multiplier is complex [22]. S-box based on residue prime numbers gives optimization in resource utilization [23]. Galois field exhibits rotational symmetry property which is used by author for S-box optimization. N-bit Boolean function masking Methodology is used for Masked AES, but design is not suitable for high throughput [24]. Author had implemented combine design of S-box and inverse S-box which enhances the area efficiency, however other transformations are not highlighted. Author had proposed balanced and pipelined architecture implementation using Virtex-6 XC6VLX240T and ASIC [25].

III. Proposed Methodology

Implementation of efficient AES is proposed in this work. Parallel pipeline architecture and optimization of S-box is done to achieve high throughput. Simulation, synthesis and implementation is done using Vivado 2017.4 and Active Aldec-HDL. For implementation FPGA virtex 5, virtex 7 is used. Lastly comparative analysis of previous work and proposed work is done considering various constraints.

A. Complete Parallel Pipeline Architecture

Making partition of circuit and placing the latches at partition edge gives pipeline of data path. Parallel processing and pipelining maximize the throughput. In proposed method each round is considered as a pipeline stage. By using registers at appropriate positions the

critical path is divided into multiple blocks. Fig. 1 shows proposed structure of one round. The registers in the architecture are utilized to store the current output of the round that is currently being run. Instead of transferring the results of each round to the subsequent, we use a register that serves as a bypass or an internal register to go around directly. Because value of the current round is kept in the register, and the following input for the round can be given as soon as the output is at its current level. And the register is used to provide the input for the following cycle, preventing the two rounds without emerging into direct interference. In the iterative looping design, this is not feasible because the next input can only be provided once the entire round-based processing is complete because the same hardware is utilized repeatedly to obtain the encrypted text. Thus, the pipelined architecture enhances the speed of execution. As first pipeline stage process is substitute byte, in our proposed work optimization of this process is done. Instead of using Galois field use of residue prime numbers for LUT is done. In Galois field based Substitute box LUT consists of 256 values whereas proposed LUT needs only 129 values. Thus reduction in resource utilization is achieved

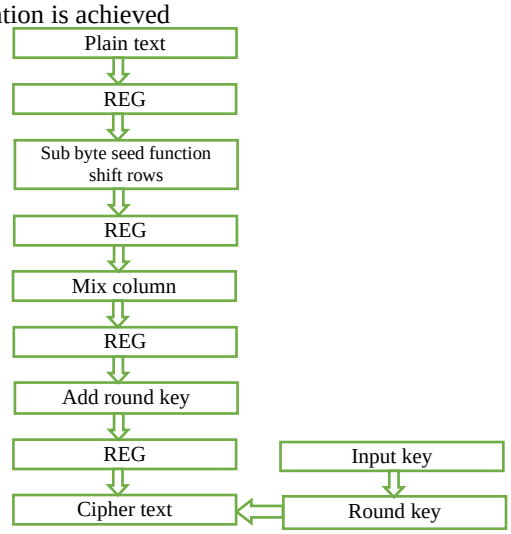

Fig. 1. Proposed pipeline structure of one round of AES.

For 128 bit AES 10 rounds are used. Each round carries modules as sub byte with seed value function, shifting of rows, mixing of columns and round key generation. The FPGA device's high-performance embedded memory blocks are employed to support the subbytes and inverse subbytes operations. To support the mix columns and inverse mix columns operations, hardware multiplication units are implemented in each data stream as shown in Fig. 1. Pipelining is created from a traditional combinational design. This is accomplished through the use of a loop unrolled architecture and the logging of intermediate data in between rounds. As data is processed in a continuous manner, the speed of execution increases in the proposed method. At each clock cycle pipelining allows to receive input and pass it to next stage of pipeline. Each pipeline round consist of substitute byte, shift rows, mix column, add round key, and key expansion operations. In first stage first operation reads data from the memory which is partitioned. After processing it is given to next stage via register. Register implementation for pipeline is as per Fig. 2.

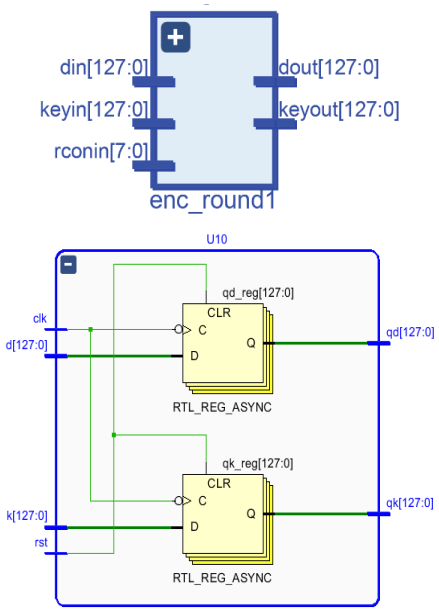

Fig. 2. RTL schematic of implemented module.

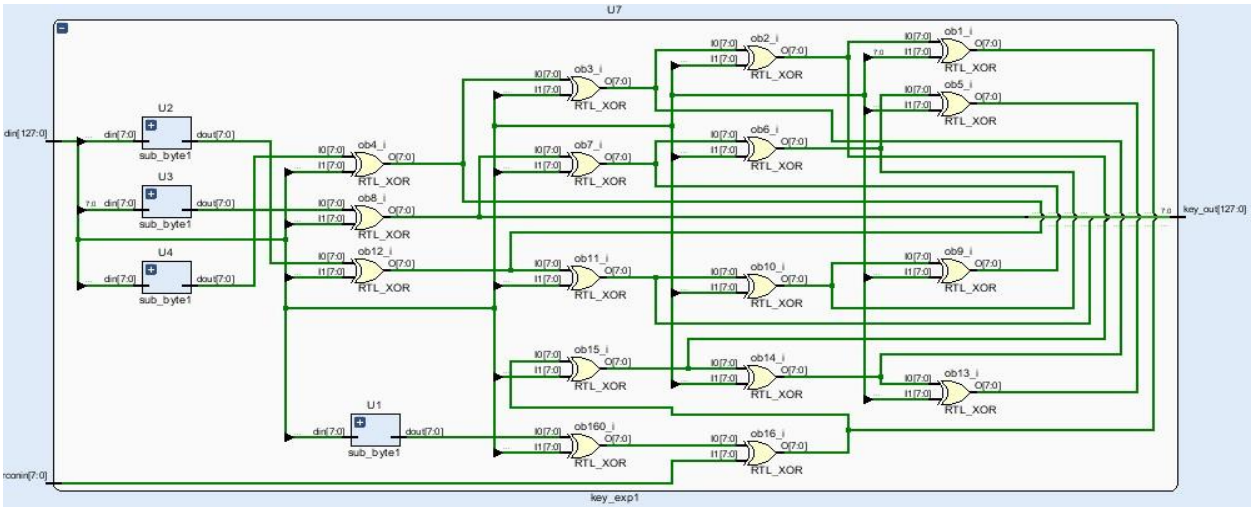

Fig. 3. Key expansion module.

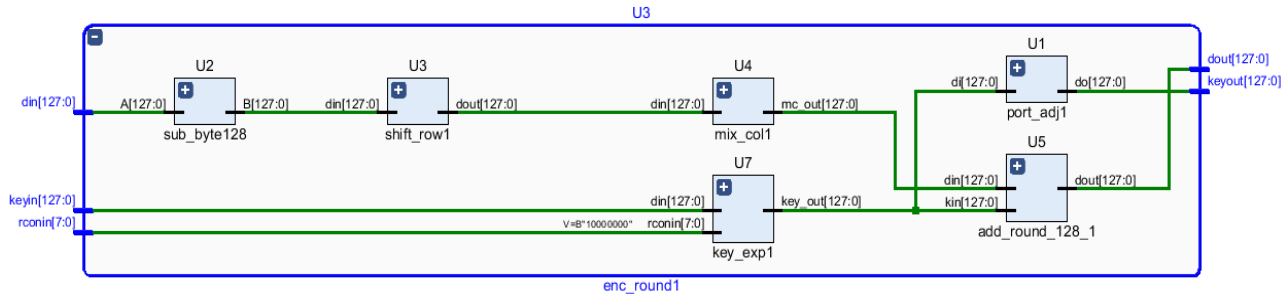

Fig. 4. Round structure of proposed method

Last stage is add round key which reads data from register and key from partitioned memory. Key expansion module implementation is as shown in Fig. 3. Add round key is the last pipeline stage that reads data from register and key from memory. RTL schematic obtained in Vivado 17.4 is shown in Fig. 2. Structure of each round obtained is depicted in Fig. 4.

B. *Efficient S-Box Using Residue Prime Numbers*

S-box is a key element of AES, which have huge impact on resource utilization. To optimize design further S-box optimization is proposed. In most of existing methods Galois field method is used for S-box. Whereas in proposed method residue prime system is used Reduction in LUTs are done by residue prime number system. For S-box record only number or its inverse is considered. It results in reduction of 12.42% LUTs as compared to previous work. It also results in less memory usage and minimal delay is required. To achieve more confusion seed value function is used on each entry of S-box.

IV. Results and Discussions

A. *Throughput and Efficiency Analysis*

FPGA implementation of proposed system is carried out using XC5v1x30ff324 and 7vx330ffg1157. Xilinx ISE design suite 14.7, Aldec HDL and Vivado 17.4 is used for completion of very large scale integration design flow. Total usage for XC5v1x30ff324 device in terms of number of slices is 1140 out of available 19200 (5%). Memory usage is 322904 KB. 7vx330ffg1157 device occupied total number of slices 2472 which leads to only 1% utilization of available slices. Simulation is as shown in Fig. 5.

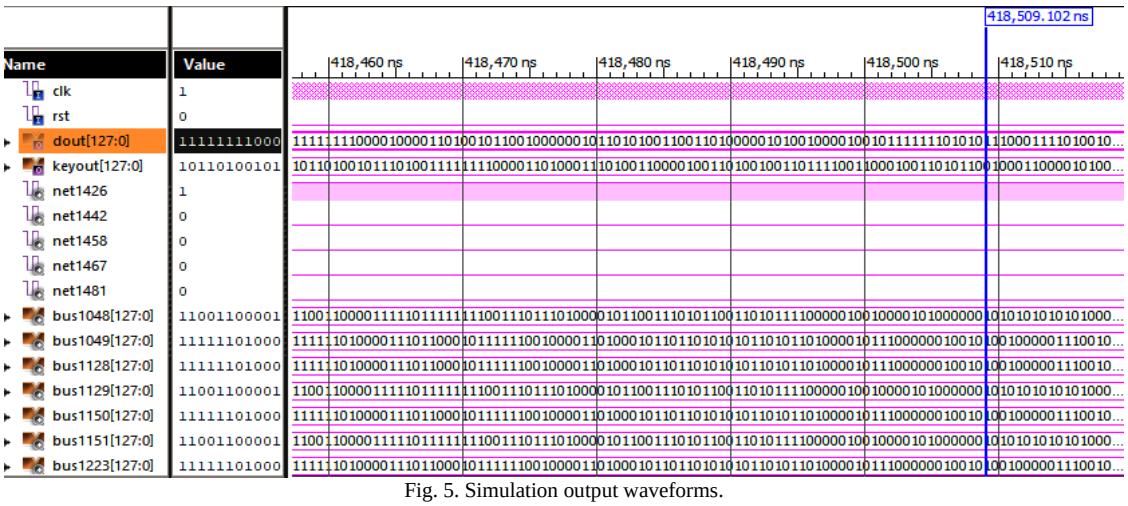

Fig. 5. Simulation output waveforms.

TABLE I: Comparative Analysis of Proposed Design with Other Existing Designs

Design	Device	Bits	Frequency (MHz)	Crit. Path (ns)	Throughput (Gbps)	Slices	Efficiency (Mbps/Slices)
[26]	XC7VH870	128	231.7	5.025	0.479	1425	0.336
[15]	XC5VLX85	128	622.4	-	79.7	5974	13.3
[27]	5AGTD3	128	-	3.88	33.015	1541	13.07
[28]	XC5VLX85	128	704.7	1.42	90.4	2940	30.74
[29]	XC5VLX330	128	533	11.74	3.8	--	-
[30]	XC6VLX240T	128	319.29	-	40.869	9071	4.51
[13]	XC6VLX240T	128	617.627	-	60.29	2252	26.77
[20]	XC5VFX70T	128	371.614	-	1.534	1604	0.959
This work	7vx330ffg1157	128	716.076	1.396	91.69	2472	37.091
This Work	XC5v1x30ff324	128	758.495	1.318	97.11	1140	85.18

Major Significant metrics for evaluation of encryption techniques are frequency, Throughput and efficiency. Throughput is calculated as (1) or (2), and (3) is used for efficiency calculation.

$$\text{Throughput} = \frac{\text{number of generated output bits}}{\text{critical path delay}} \quad (1)$$

$$\text{Throughput} = \frac{\text{output bits}}{1/\text{maximum frequency}} \quad (2)$$

$$\text{Efficiency} = \frac{\text{throughput of design}}{\text{total area(number of slices)}} \quad (3)$$

For proposed design number of generated output bits is 128. Throughput and efficiency achieved for XC5v1x30ff324 device is 97.11 and 85.18 respectively. For 7vx330ffg1157 device throughput and efficiency obtained is 91.69 and 37.091 respectively. Table I gives comparative of proposed design with other existing designs based on synthesis results. As per Fig. 5 comparative analysis shows that our proposed design archives highest throughput and efficiency. Due to pipeline structure and optimized S-box design it is possible to achieve better performance. Constrained nodes are used in internet of things. Such nodes mainly concern with less resource utilization and critical path delay. As per analysis shown in Table I, proposed method gives minimum critical delay and less slice usage. Hence proposed method is suitable for constrained nodes in IoT systems.

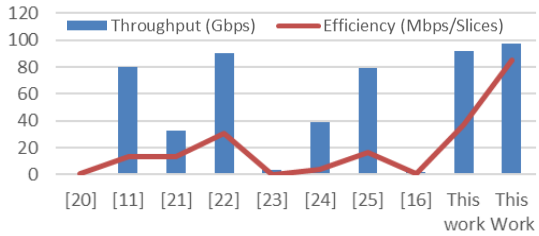

Fig. 5. Performance based on throughput and efficiency.

B. Avalanche Effect Analysis

Performance analysis of AES is done based on avalanche effect and Strict Avalanche Criterion (SAC). Avalanche effect finds diffusion and confusion property. It is calculated as change in output for one bit change in input. If avalanche effect is more than 50% then it satisfies the strict avalanche criterion.

$$\text{Avalanche effect} = \frac{\text{number of flipped bits}}{\text{total number of bits}} \quad (4)$$

Table II gives avalanche test result for change of 1 bit in plaintext input while key kept constant. It shows that average avalanche effect of proposed method is 58%. Hence proposed method is exhibiting strong characteristics to prevent linear and differential attacks and significantly satisfies strict avalanche criterion.

C. National Institute of Standard and Technology (NIST) Test Validation by Runs Test, Frequency Test and Non-Overlapping Template Matching Test

Frequency test: This test finds closeness of bits. In a data block of N bit it measures number of zeros and ones. If frequency is approximately $N/2$ then it satisfies frequency test. In proposed method block size is 128 bits. Total count of ones and zeros are 65 and 63 respectively. Thus it passes frequency test

Runs test: This test verifies, whether the runs of 1's and 0's of different lengths are within the acceptable range as set by NIST for any random sequence. Probability of obtaining results P-value determines randomness of output values. P-value is probability of obtaining a test statistic as larger than the one observed if the sequence is random. Comparison of P-value with significance value of α is done. Significance value 0.05 is considered satisfactory by NIST. Statistical analysis for run test is given in Table III.

TABLE III: STATISTICAL ANALYSIS OF RUN TEST

No of bits	Sample Mean	Number of Observations		Number of Runs		P-value
N	K	$\leq K$	$>K$	Observed	Expected	0.891
128	0.554688	57	71	65	64.23	

Null hypothesis (H_0) is the order of the data is random while alternative hypothesis (H_1) is the order of the data is not random. If P-value is less than or equal to α then the order of the data is not random (reject H_0). The decision is to reject the null hypothesis and conclude that the order of the data is not random. Alternatively if the P-value is greater than the significance level, the decision is to fail to reject the null hypothesis which cannot conclude the order of the data is not random. In proposed method P-value obtained is 0.891, which clearly shows that randomness is observed in output data.

TABLE II: AVALANCHE TEST RESULTS

Test	Key	Plaintext	Ciphertext	Avalanche effect
1	0f470caf15d9b77f71e8ad67c959d698	0189fe7623abdc5445cdba3267ef9810	ff0869640b53341484bfab8f4a7c43b9	55.46
		0189fe7623abdc5445cdba3467ef9810	c3cb1e8d4bbd2d415f03c9c63dce3f62	
2	0f470caf15d9b77f71e8ad67c959d698	0189fe7623abdc5445cdba3267ef9810	ff0869640b53341484bfab8f4a7c43b9	54.68
		0189fe7623abdc5445cdba3267ef9010	278da302d4fdd6acce1506266e0ca456	
3	0f470caf15d9b77f71e8ad67c959d698	0189fe7623abdc5445cdba3267ef9810	ff0869640b53341484bfab8f4a7c43b9	53.125
		0189fe7623abdc5445cdba3267ef1810	ebcab49637969b316a68a998e040fd80	
4	0f470caf15d9b77f71e8ad67c959d698	0189fe7623abdc5445cdba3262ef9810	fb63cc8eb3cb735e61fae2a12e73e5d1	60.156
		0189fe7623abdc5405cdba3267ef9810	7c872c3aca27b67da6d1aafc9335182f	
5	0f470caf15d9b77f71e8ad67c959d698	0189fe7623abdc5445cdba3262ef9810	fb63cc8eb3cb735e61fae2a12e73e5d1	54.68
		0189fe7623abdc0445cdba3267ef9810	3f2e4c97ee7c1a11057ce9775a3509cc	

Non-overlapping template matching test: occurrences of non-periodic pattern is detected by this test. For searching N bit Patten n bit window is used. P-value is calculated as per below equations

$$\mu = \frac{(N-n+1)}{2^n} \tag{5}$$

$$\delta = N\left(\frac{1}{2^n} - \frac{2n-1}{2^{2n}}\right) \tag{6}$$

$$\gamma^2 = \frac{w-\mu}{\delta} \tag{7}$$

$$p\text{-value} = i_{\text{gmac}}\left[\frac{M}{2}, \frac{\gamma^2}{2}\right] \tag{8}$$

where μ is the mean, N is the length of block, n is the Window size, δ is the reference distribution, I_{gmac} is the incomplete gamma function, and w is the matching weights. For calculation of *P*-value based on above equations, different block sizes and window sizes are considered for this test. Average *P*-value obtained for proposed method is 0.257. As obtained *P*-value is greater than 0.01, it states that data is random in nature.

Overlapping template matching test: occurrence of n-bit pattern is calculated in this test. Different output pattern of proposed method is tested to find *P-value*. The search for matches proceeds by generating m-bit window on the sequence, comparing the bits within that window against m bit template for matching and incrementing a counter when there is a match. *P-value* obtained for this test is 0.51. As it is more than 0.01, it clearly indicates that randomness is observed in the output data.

V. Conclusion

Proposed AES based on pipeline structure and optimized S-box provides high throughput and efficient encryption technique. FPGA implementation proved that less area is used and delay is also reduced significantly, which makes this system valuable for constrained nodes. Effectiveness of design is evaluated using avalanche effect and SAC. Significant increase in avalanche effect is obtained and strict avalanche criterion is achieved effectively. Design is validated using NIST validation tests, which proves that randomness factor is achieved. Resource utilization is also reduced as for 7vx330ffg1157 only 1% number of slices from available slices are used. For XC5v1x30ff324 device throughput of is 97.11Gbps and efficiency of 85.18 Mbps/slice is achieved with maximum frequency of 758.495 MHz. We plan to extend our work for ASIC synthesis flow with Cryptanalysis performance.

Conflict of Intrest

The authors declare no conflict of interest.

Author Contributions

Both authors have analyzed the complete work and prepared manuscript. Both author had approved final copy.

References

[1] A. Soltani and S. Sharifian, "An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA," *Microprocess. Microsyst.*, vol. 39, no. 7, pp. 480–493, 2015.

[2] D. D. Diambeki, R. E. Mandiya, K. Kyamakya, and S. K. Kasereka, "ScienceDirect securing the light escaping in a Li-Fi network environment," *Procedia Comput. Sci.*, vol. 201, pp. 684–689, 2022.

[3] A. Iqbal and T. Iqbal, "Low-cost and secure communication system for remote micro-grids using AES cryptography on ESP32 with LoRa module," in *Proc. IEEE Electr. Power Energy Conf.*, 2018.

[4] L. R and K. M, "Enhancing the security of AES through small scale confusion operations for data communication," *Microprocess. Microsyst.*, vol. 75, 2020.

[5] H. R. Moorthy, V. Bangera, Z. Amrin, J. N. Avinash, and K. N. S. Rao, "WSN in defence field: A security overview," in *Proc. 4th Int. Conf. IoT Soc. Mobile, Anal. Cloud, ISMAC 2020*, 2020, pp. 258–264.

[6] T. Hidayat, D. S. T. Franky, and R. Mahardiko, "Forecast analysis of research chance on AES algorithm to encrypt during data transmission on cloud computing," in *Proc. 2nd Int. Conf. Broadband Commun. Wirel. Sensors Powering, BCWSP 2020*, 2020, pp. 163–166.

[7] I. A. Awan, M. Shiraz, M. U. Hashmi, Q. Shaheen, R. Akhtar, and A. Ditta, "Secure framework enhancing AES algorithm in cloud computing," *Secur. Commun. Networks*, vol. 2020, 2020.

[8] D. Evangelin, R. Venkatesan, K. Ramalakshmi, S. Cornelia, and J. Padmhavathi, "Survey in finding the best algorithm for data analysis of privacy preservation in healthcare," *Lect. Notes Data Eng. Commun. Technol.*, vol. 35, no. August 2020, pp. 743–746, 2020.

[9] H. K. Kim and M. H. Sunwoo, "Low power AES using 8-Bit and 32-Bit datapath optimization for small Internet-of-Things (IoT)," *J. Signal Process. Syst.*, vol. 91, no. 11–12, pp. 1283–1289, 2019.

[10] M. Qasaimeh, R. S. Al-Qassas, and M. Ababneh, "Software design and experimental evaluation of a reduced aes for iot applications," *Futur. Internet*, vol. 13, no. 11, pp. 1–21, 2021.

[11] H. Zodpe and A. Sapkal, "FPGA-Based high-performance computing platform for cryptanalysis of AES algorithm," *Adv. Intell. Syst. Comput.*, vol. 1025, September, pp. 637–646, 2020.

[12] K. S. Dhanalakshmi and R. A. Padmavathi, "A survey on VLSI implementation of AES algorithm with dynamic S-Box," *J. Appl. Secur. Res.*, pp. 1–15, 2021.

[13] X. Zhang, M. Li, and J. Hu, "Optimization and implementation of AES algorithm based on FPGA," in *Proc. IEEE 4th Int. Conf. Comput. Commun. ICCC 2018*, 2018, pp. 2704–2709.

[14] T. M. Kumar, K. S. Reddy, S. Rinaldi, B. D. Parameshachari, and K. Arunachalam, "A low area high speed fpga implementation of aes architecture for cryptography application," *Electron.*, vol. 10, no. 16, 2021.

[15] K. Shahbazi and S. B. Ko, "High throughput and area-efficient FPGA implementation of AES for high-traffic applications," *IET Comput. Digit. Tech.*, vol. 14, no. 6, pp. 344–352, 2020.

[16] S. S. Priya, P. Karthigaikumar, N. M. Siva Mangai, and P. K. G. Das, "An efficient hardware architecture for high throughput AES encryptor using MUX based sub pipelined S-Box," *Wirel. Pers. Commun.*, vol. 94, no. 4, pp. 2259–2273, 2017.

[17] K. Shahbazi, S. Ko, and S. Member, "Area-Efficient Nano-AES implementation for internet-of-things devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1–13, 2020.

[18] T. M. Kumar and P. Karthigaikumar, "FPGA implementation of an optimized key expansion module of AES algorithm for secure transmission of personal ECG signals," *Des. Autom. Embed. Syst.*, vol. 22, no. 1–2, pp. 13–24, 2018.

[19] 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.

[20] M. Bedoui, H. Mestiri, B. Bouallegue, B. Hamdi, and M. Machhout, "An improvement of both security and reliability for AES implementations," *J. King Saud Univ. - Comput. Inf. Sci.*, 2022.

[21] A. Shrivastava, D. Haripriya, Y. D. Borole, A. Nanoty, C. Singh, and D. Chauhan, "High performance FPGA based secured hardware model for IoT devices," *Int. J. Syst. Assur. Eng. Manag.*, 2022.

[22] C. A. Murugan, P. Karthigaikumar, and S. S. Priya, "FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-box techniques for compact applications," *Automatika*, vol. 61, no. 4, pp. 682–693, 2020.

[23] S. Sanap, V. More, S. Cience, and T. Echnology, "Design of efficient S-box for advanced encryption standard," *Journal of Integrated Science and Technology*, vol. 10, no. 1, pp. 39–43, 2022.
[24] F. Wegener, L. De Meyer, and A. Moradi, *Spin Me Right Round Rotational Symmetry for FPGA-Specific AES: Extended Version*, vol. 33, no. 3. Springer US, 2020.
[25] Y. T. Teng, W. L. Chin, D. K. Chang, P. Y. Chen, and P. W. Chen, "VLSI architecture of S-Box with high area efficiency based on composite field arithmetic," *IEEE Access*, vol. 10, pp. 2721–2728, 2022.
[26] I. Algredo-Badillo, K. A. Ramírez-Gutiérrez, L. A. Morales-Rosales, D. P. Bautista, and C. Feregrino-Uribe, "Hybrid pipeline hardware architecture based on error detection and correction for aes," *Sensors*, vol. 21, no. 16, pp. 1–26, 2021.
[27] R. Mondal, H. Ngo, J. Shey, R. Rakvic, O. Walker, and D. Brown, "Efficient architecture design for the AES-128 algorithm on embedded systems," in *Proc. 17th ACM Int. Conf. Comput. Front. 2020, CF 2020 - Proc.*, 2020, pp. 89–97.
[28] M. B. Chellam and R. Natarajan, "AES hardware accelerator on FPGA with improved throughput and resource efficiency," *Arab. J. Sci. Eng.*, vol. 43, no. 12, pp. 6873–6890, 2018.
[29] H. Lee, Y. Paik, J. Jun, Y. Han, and S. W. Kim, "High-throughput low-area design of AES using constant binary matrix-vector multiplication," *Microprocess. Microsyst.*, vol. 47, pp. 360–368, 2016.
[30] Y. Wang and Y. Ha, "FPGA-based 40.9-gbits/s masked AES with area optimization for storage area network," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 60, no. 1, pp. 36–40, 2013.

Copyright © 2022 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

Sarita Sanap is currently working as assistant professor in Department of Electronics and Telecommunication Engineering at Maharashtra Institute of Technology, Aurangabad (MS), India. She is pursuing Ph.D. degree from National Institute of Electronics and Information Technology, Dr.B.A.Marathwada University, India. She has 14 years of experience in teaching. Her area of interest are Information Security and VLSI.

Dr. Vijayshree More is currently working as associate professor in department of Electronics and Computer Engineering department at Jawaharlal Nehru Engineering College, MGM University, Aurangabad (MS), India. She has 34 years of experience in teaching. She has more than 18 publications in reputed journals and conferences. Her area of interest are embedded systems and image processing.