Performance Evaluation of Lightweight Advanced Encryption Standard Hardware Implementation

Herman B. Acla, Bobby D. Gerardo

Abstract: Advanced Encryption Standard (AES) is one of the most secured encryption algorithm because of its robustness and complexity. Because of its complexity, AES has slow computation. This paper presents a Lightweight Advanced Encryption Standard (LAES) design by replacing the MixColumn transformation of the traditional AES with a 128-bit permutation to lessen its computational complexity. Implementation of hardware cryptographic encryption aims to find the best trade-off between throughput and resource utilization. The proposed design is synthesized on various Field Programmable Gate Array (FPGA) devices and achieves the maximum clock frequency of 480.50 MHz with the highest throughput of 6.15 Gbps when synthesized on Virtex 7 XC7VX690T. The results on other devices show a higher throughput, better performance efficiency, and lesser area utilization when compared to the existing AES hardware implementation.

Index Terms*: *AES algorithm, FPGA, Hardware based encryption, Permutation table.

I. INTRODUCTION

Internet of things (IoT), is a concept that describes the paradigm of internetworking electronic devices, to be connected to the internet and communicate with each other, to exchange and collect data through a wireless medium. These electronic devices are equipped with sensors, software, electronics, and network connectivity that enables to identify and communicate themselves with other devices. IoT is showing its potential in economic and social application by providing real time data communication in multipurpose application. These are made possible through the use of Radio Frequency Identifier (RFID), cloud services, QR codes, Wireless Sensor Networks (WSNs), and other portable technologies [1]. Consequently, these technologies become a critical component in various IoT application domains including Smart Cities, Smart Grids, Connected Car, Connected Industry, Smart Farming, Connected Health, Nano-scale applications, and many other fields [2],[3]. WSNs utilize portable and low-cost sensor which provide better support for IoT applications, since deployment cost is crucial in IoT application [4]. But this portability also draws disadvantages as stated in the work of [5] and [6] that the sensor nodes suffers in storage capacity, power and energy resources constraints. Further, [7] and [8] pointed out that nodes have limited computational and memory resources.

Communication between nodes and node to base stations (BS) is susceptible to network attacks. Keeping the confidentiality, integrity, and accuracy of the data transmitted by sensor nodes to its destination is very important. Thus, security countermeasures should be implemented. But the compact design of sensor nodes with little security attributes increases security challenges and vulnerabilities. Further, due to resources contraints, security features is considered last in the design of WSN [9], [10]. Given the resource limitation and salient feature of WSN, the addition of security is the last thing considered in the design of WSN for the reason that it produces unwanted surplus to energy cost and processing overload cost to sensor nodes [11]. Lower performance of Field Programmable Gate Array (FPGA) – based WSN nodes that uses cryptographic encryption is influenced by the complexity of the security protocol [12]. Performance metrics of hardware cryptographic encryption implementation are area, timing, power, throughput, and performance efficiency [13]. Advanced Encryption Standard (AES) is considerably one of the highly implemented symmetric cryptography today and most dependable encryption in terms of security which can be directly associated to the robustness and the complexity of the algorithm [14], [15]. This study will address the issue in the performance of secured FPGA-based WSN node by optimizing the AES cipher through hardware encryption. The main contribution of the study is the development of high throughput, lesser area, and high performance efficiency hardware based lightweight encryption algorithm.

II. RELATED LITERATURE

Implementation of AES algorithm can be categorized as software and hardware implementation. Significant amount of research has been done on the comparison of hardware and software implementation of AES algorithm. According to [16], for security of WSNs, hardware implementation is more efficient than software in terms of time complexity, throughput and power consumed during the security process. In the paper of [17] and [18], it was established that when it comes to real time high speed applications, hardware implementation of AES is more suitable than software implementation.

Revised Manuscript Received on 30 July 2019.
* Correspondence Author

Herman B. Acla, Graduate School, Technological Institute of the Philippines, Quezon City, Philippines and Northern Iloilo Polytechnic State College, Iloilo, Philippines.
Bobby D. Gerardo, Institute of Information and Communications Technology, West Visayas State University, Iloilo City, Philippines

In the work of [11], they compared the performance of AES in three different implementations on WSN.

The original AES, table look up AES and hardware AES. Result have shown that on encrypting data, hardware implementation of AES is the fastest; further in the work of [19] and [20], the result shows that hardware implementation of AES has a better throughput and more energy efficient. Similarly in the work of [21], when compared to AES software implementation it was justified by the result of their experiment that AES hardware implementation has lesser energy consumption and performs faster.

Significant amount of work has been done on hardware-based implementation of AES algorithm on FPGA. The objective of these researches is to design and implement the AES algorithm and improve its operating frequency, enhance the throughput, increase the efficiency, lessen the area utilization, lower the latency, and reduce the power dissipation.

In order to achieve a faster encryption/decryption of data, [22] implemented pipelined architecture on Xilinx Virtex 5. Pipelined architecture enables the processing of multiple blocks simultaneously or in parallel. This design enhances the throughput at the cost of the increase in area utilization since registers are used to store the results of the individual stages of the design which add-ons to the logic resources used by the design [22], [23].

The work of [24] proposed a high-throughput masked AES using look up table (LUT)-based masked S-box. Given that most the masked AES implementation is using unrolled architecture, which utilizes extremely large quantity of FPGA resources, the proposed design aimed to optimize the area utilization for a masked AES using unrolling technique. The proposed design was made possible through mapping the AES operations from Galois Field (GF) 2^8 to GF 2^4.

In the paper of [25], a modified S-box and an improved key generation for initial key required for encryption using PN Sequence Generator was introduced. The developed AES with proposed modifications resulted to a significant improvement in throughput after the design is synthesized on various FPGA devices.

Several researches exploited the AES protocol to streamline the extent of the code and improve its performance. In the work of [26], multiple S-boxes were used and MixColumn was replaced by a different S-box. The implementation of the modified AES resulted to a better performance compared to the traditional AES in terms of speed and security. In the paper of [27], used the GF(2^4) instead of GF(2^8) to get the irreducible polynomials as basis for producing the S-box. In the paper of [28], they proposed a modified AES by using a "key" such as timestamp during transmission, node name, or last two bits of MAC address to be EX-ORed to the value of after it was substituted using the S-box.

III. OVERVIEW OF AES ALGORITHM

AES encryption algorithm is a block cipher developed by the Belgian cryptographers Joan Daeman and Vincent Rijmen and was adopted by National Institute of Standards and Technology (NIST) as a replacement for the much slower and small block size Data Encryption Standard (DES). AES is a symmetric block cipher with a block length of 128 bits, uses encryption key and several rounds of encryption depending on the length of the key. The key can be 128, 192 and 256 bits and the number of rounds with respect to key length is 10, 12, and 14 respectively [29]. Generally, these referred to as AES-128, AES-192 and AES-256 based on the key length used. Round function operates using a set of four operations on a two dimensional 4 X 4 array of bytes called the States. Substitute bytes – a byte substitution that utilizes an S-box, Shift Rows – a permutation, Mix Columns – a linear diffusion, and Add Round Key – a bitwise XOR operation of the current block state with a portion of the current expanded key. The over-all structure of AES128 is shown in Fig. 1.

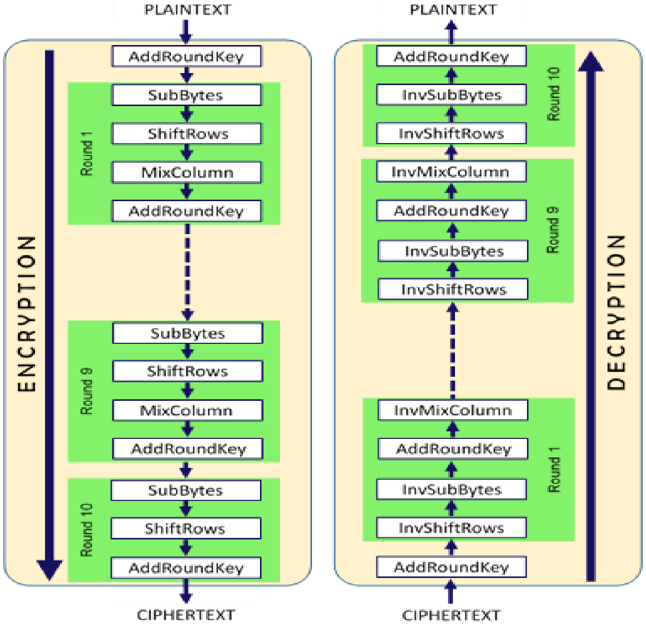


Fig. 1: AES Encryption/Decryption Structure

A. SubBytes Transformation

SubBytes, or substitute byte transformation is a simple table lookup. AES defines S-box as a 16 x 16 matrix of byte values that contains a permutation of all possible 256 8-bit values. The SubBytes transformation substitute each individual byte from the State with the corresponding value from the S-Box. The leftmost hexadecimal value of the byte corresponds to a row value and the rightmost hexadecimal value of the byte corresponds to a column value. To substitute the unique byte output value, these row and column values serve as indexes into the S-box. Fig.2 shows the SubBytes operation. The InvSubBytes or inverse substitute byte transformation utilizes the inverse S-box.

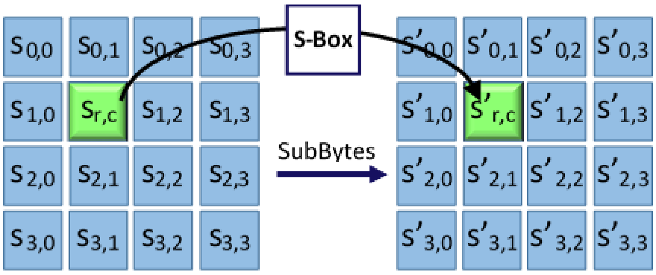

$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{r,c}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
S-Box
SubBytes
$S'_{0,0}$ $S'_{0,1}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{r,c}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,1}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,1}$ $S'_{3,2}$ $S'_{3,3}$

Fig. 2: Substitute Byte Transformation

B. ShiftRows Transformation

ShiftRows or shift row transformation, is shown in Fig.3. Each row of the State matrix is shifted to the left cyclically depending on the row index. The first row of State remains unchanged. For the second row, one byte position is shifted to the left of the matrix. For the third row, two bytes position is shifted to the left of the matrix. For the fourth row, three bytes position is shifted to the left of the matrix. InvShiftRows or the inverse shift row transformation, cyclically shifts the last three rows in the opposite direction of the state matrix, with a one-byte shift cyclically to right for the second row, two-bytes is shifted to right in the third row and three-bytes is shifted to the right in the fourth row.

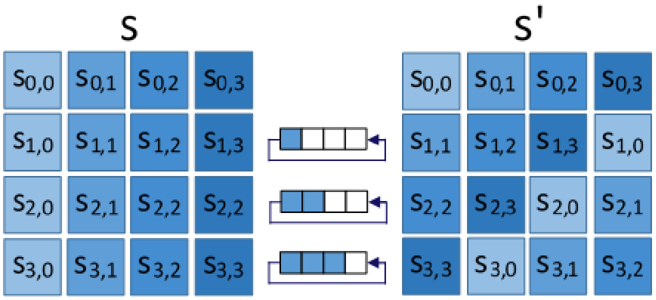


Fig. 3: Shift Row Transformation

C. MixColumn Transformation

MixColumn or mix column transformation is shown in Fig.4. It is a linear diffusion process that operates on each of the state matrix column individually. The transformation can characterized in terms of polynomial arithmetic where every column of the state matrix is considered as four term polynomial and the individual multiplication and addition is performed over GF (2^8).

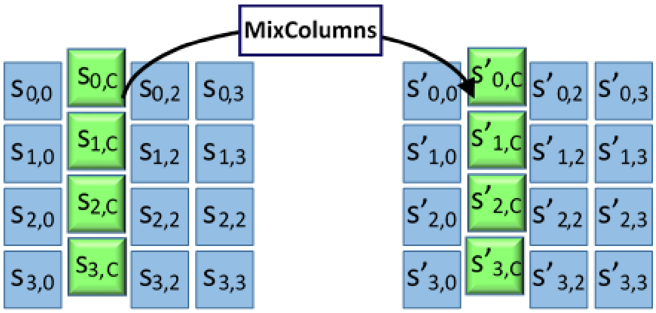


Fig. 4: Mix Column Transformation

D. AddRoundKey Transformation

AddRoundKey or forward add round key transformation, is a simple bitwise XOR operation between the 128 bits of State and the 128 bits of the round key. As presented in Fig.5, the transformation is considered as a columnwise operation between the State column and the corresponding column of the round key. Given that the function of XOR is its own inverse, the operation of the inverse add round key transformation is the same with the forward add round key transformation.

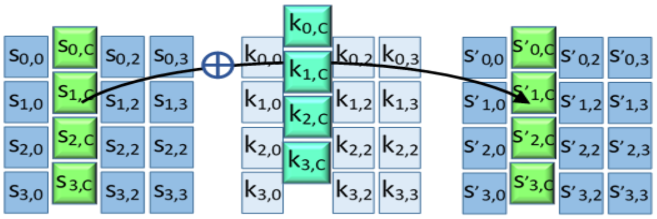


Fig. 5: Add Round Key Transformation

IV. PROPOSED LIGHTWEIGHT AES ALGORITHM

The main focus of this study is the development of Lightweight Advanced Encryption Standard (LAES) algorithm through hardware based implementation of its lightweight modification. VHDL programming will be used in developing the lightweight cipher based on AES. Shown in Fig. 6, is the block diagram of the proposed algorithm. The encryption sequence of the algorithm is a replication of traditional AES with a modification of substituting a 128-bit Permutation to the Mix Column transformation. Encryption process of the LAES starts with the first round key is XORed to the plain text followed by nine rounds sequenced as:

- Substitute bytes (SubBytes)
- Shift rows (ShiftRows)
- Permutation
- Add round key(AddRoundKey)

The tenth and the last round in order to produce the Cipher Text, follows the same sequence except that the Permutation step is omitted.

Decryption process of LAES begins with AddRoundKey to the Cipher Text, then nine iterations of the following:

- Inverse shift rows (InvShiftRows)
- Substitute bytes (SubBytes)
- Inverse permutation (InvPermutation)
- Add round key(AddRoundKey)

The tenth and the last round in order to produce the Plaintext, follows the same sequence except that the InvPermutation step is omitted.

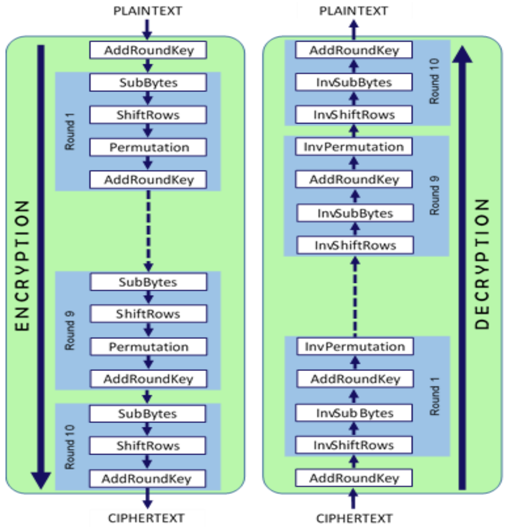


Fig. 6: Proposed LAES Encryption/Decryption Structure

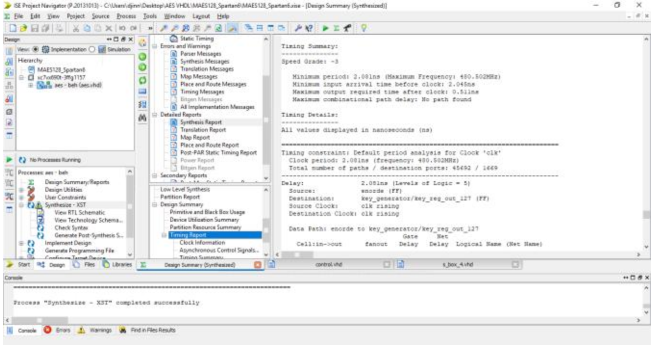
The reason for the substitution is that MixColumn process involves complex and numerous calculation thus, directly affects the area and performance of the algorithm during encryption and decryption. Thus, the introduction of an alternative nonlinear process with lesser computational complexity is assumed to improve the operating frequency, increase the throughput, lessen the area utilization and improve the performance efficiency. The 128 bit permutation table is shown in Table I.

Table I: LAES Permutation Table

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	122	114	106	98	90	82	74	66	58	50	42	34	26	18	10	2
2	124	116	108	100	92	84	76	68	60	52	44	36	28	20	12	4
3	126	118	110	102	94	86	78	70	62	54	46	38	30	22	14	6
4	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8
5	121	113	105	97	89	81	73	65	57	49	41	33	25	17	9	1
6	123	115	107	99	91	83	75	67	59	51	43	35	27	19	11	3
7	125	117	109	101	93	85	77	69	61	53	45	37	29	21	13	5
8	127	119	111	103	95	87	79	71	63	55	47	39	31	23	15	7

V. DATA COLLECTION, ANALYSIS AND DISCUSSION

The proposed lightweight algorithm is developed using Very High Speed Integrated Circuit Hardware Description Language (VHDL) programming. A language for describing the structure and behavior of electronic systems such as FPGA and integrated circuits. Using various FPGA devices, the proposed algorithm is synthesized. A screen capture of a synthesis report is shown in Fig. 7. The result of the synthesis on various devices is presented in Table II.



```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 2.081ns (Maximum Frequency: 480.502MHz)
   Minimum input arrival time before clock: 2.045ns
   Maximum output required time after clock: 0.511ns
   Maximum combinational path delay: No path found

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 2.081ns (frequency: 480.502MHz)
  Total number of paths / destination ports: 45692 / 1669
-------------------------------------------------------------------------
Delay:               2.081ns (Levels of Logic = 5)
  Source:            encrde (FF)
  Destination:       key_generator/key_reg_out_127 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: encrde to key_generator/key_reg_out_127
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
```

```
=========================================================================

Process "Synthesize - XST" completed successfully
```

Fig. 7: Synthesis report of XC7VX690T

The parameters for evaluating the performance of the proposed lightweight AES are area, throughput, and performance efficiency.

Area utilization for FPGAs is identified with respect to number of slices [20], [25]. Throughput signifies the number of bits processed per unit time and is specified in Gbps or Mbps. Throughput can be expressed as

$$Throughput = \frac{No.of\ bits\ processed * Fmax}{Total\ clock\ cycles} \quad (1)$$

where the No. of bits processed is 128, Fmax refers to the maximum frequency shown in the synthesis report and Total clock cycles refers to the number of cycles per encryption block. Performance efficiency is defined as the ratio of the throughput over the area [20] and expressed as

$$Efficiency = \frac{\text{Throughput}}{\text{Area}} \quad (2)$$

The proposed LAES reported the maximum clock frequency of 480.50 MHz with the highest throughput of 6.15 Gbps on Virtex 7 XC7VX690T device. It also attained the highest performance efficiency of 7.18 Mbps/Slice on the same device utilizing 857 slices equivalent to 0.0989% of the total available slices.

Table II: Resource Utilization of Proposed LAES on various FPGA Devices

Device	No. of Slices	Fmax (MHz)	Throughput (Gbps)	T/S Mbps/Slice
XC7VX690T	857	480.50	6.15	7.18
XC6VLX240T	858	445.01	5.70	6.64
XC6SLX150	867	199.17	2.55	2.94
XC5VLX110T	866	334.10	4.28	4.94
XC5VSX240T	866	313.68	4.02	4.64
XC5VLX50	866	334.10	4.28	4.94
XC4VLX60	3556	228.99	2.93	0.82

Comparisons of the result of synthesis with existing non-pipelined implementations is shown in Table III. In the XC7VX690T device, the proposed LAES is 78.79% better in terms of throughput and an improvement of 1.37% in performance efficiency compared to [30] although the proposed design has a higher area by 76.34% in slice usage. Similarly compared to [25], although there is an increase in area by 22.96% in slice utilization, there is an improvement of 41.71% in throughput and 15.26% in performance efficiency. In XC6VLX240T device, the proposed LAES improved the throughput by 202.98% and 65.11% compared to [24] and [30] respectively. Although the throughput in [25] is higher by 3.94%, the proposed LAES is better in terms of performance efficiency by 361.3% and has a lesser in area utilization by 79.05%. The throughput in [25] is higher using XC6SLX150 but in terms of area and performance efficiency, the proposed LAES is better by 84.42% and 444.53% respectively. In XC4VLX60 device, there is an increase in throughput by 42.28% and 6.97% compared to [30] and [25] respectively.

Table III: Comparison of LAES Results with Existing Non-Pipelined AES Architecture Designs

Design	Device	No. of Slices	Fmax (MHz)	Throughput (Gbps)	T/S Mbps/Slice
[25]	XC7VX690T	697	372.98	4.34	6.22
[30]	XC7VX690T	486	322.58	3.44	7.08
Proposed LAES	XC7VX690T	857	480.50	6.15	7.18
[24]	XC6VLX240T	15612	14.69	1.88	0.12
[25]	XC6VLX240T	4095	463.42	5.93	1.44
[30]	XC6VLX240T	335	323.73	3.45	10.29
Proposed LAES	XC6VLX240T	858	445.01	5.70	6.64
[25]	XC6SLX150	5566	237.45	3.03	0.54
Proposed LAES	XC6SLX150	867	199.17	2.55	2.94
[25]	XC4VLX60	20818	214.48	2.74	0.13
[30]	XC4VLX60	1975	192.68	2.06	1.04
Proposed LAES	XC4VLX60	3556	228.99	2.93	0.82

The comparison of existing pipelined implementation to various FPGA devices to the synthesis result of proposed LAES is presented in Table IV. For the purpose of comparison, the computation of throughput, is based on the works of [22], [23], [25] and [31] where latency is not considered. Comparing the synthesis report to the implementation of [22], [23], and [25] in XC5VLX110T shows that the proposed LAES increases the throughput by 22.57%, 65.18% and 43.84% respectively.

Further, the proposed LAES improved the performance efficiency by 1596.98%, 529.87% and 1551.57% when compared to [23], [25] and [22]. In the same device, the proposed LAES has lesser area utilization by 90.26%, 77.14%, and 92.58% as compared to [23], [25] and [22] respectively. Similarly, in devices XC5VSX240T and XC5VLX50, the proposed design is lesser in terms of area utilization by 74.68% and 87.15%, an improvement in throughput by 57.46% and 13.43%, better in performance efficiency by 522.33% and 780.25% when compared to the pipelined implementation of [25] and [31] respectively.

Table IV: Comparison of LAES Results with Existing Pipelined AES Architecture Designs

Design	Device	No. of Slices	Fmax (MHz)	Throughput (Gbps)	T/S Mbps/Slice
[22]	XC5VLX110T	11677	272.59	34.89	2.99
[23]	XC5VLX110T	8896	202.26	25.89	2.91
[25]	XC5VLX110T	3788	232.30	29.73	7.84
Proposed LAES	XC5VLX110T	866	334.10	42.76	49.38
[25]	XC5VSX240T	3420	199.18	25.50	7.45
Proposed LAES	XC5VSX240T	866	313.68	40.15	46.36
[31]	XC5VLX50	6741	294.80	37.70	5.61
Proposed LAES	XC5VLX50	866	334.10	42.76	49.38

VI. CONCLUSION AND FUTURE WORKS

Hardware implementation of encryption/decryption algorithm should have a good trade-off between area utilization, throughput, and efficiency. A hardware based lightweight AES design aimed to enhance the throughput, lessen the area utilization increase the performance efficiency of the algorithm was presented in this paper. The design streamlined the AES128 structure by replacing MixColumn transformation with 128 bit Permutation. The proposed LAES was synthesized on various FPGA devices and compared the throughput, area utilization, and performance efficiency to the existing AES hardware based implementation. Results have shown that the proposed LAES achieved better throughput and performance efficiency without sacrificing the area utilization.

The proposed LAES achieved the highest clock frequency of 480.50 MHz with the highest throughput of 6.15 Gbps when synthesized on Virtex 7 XC7VX690T. The performance efficiency is 7.18 Mbps/Slice utilizing 857 slices equivalent to 0.0989% of the total available slices on the device. Future works may focus on other modes of operation and other architecture of AES. ASIC implementation should also be considered.

ACKNOWLEDGMENT

H. Acla would like to thank Northern Iloilo Polytechnic State College and the Commission on Higher Education for their support in this endeavour.

REFERENCES

1. I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises", Business Horizons, vol. 58, no. 4, pp. 431-440, 2015. Available: 10.1016/j.bushor.2015.03.008.
2. G. Gardašević et al., "The IoT Architectural Framework, Design Issues and Application Domains", Wireless Personal Communications, vol. 92, no. 1, pp. 127-148, 2016. Available: 10.1007/s11277-016-3842-3.
3. N. Lohana and M. M. Roja, "A Review of the Internet of Things", International Journal of Recent Technology and Engineering (IJRTE), vol. 5, no. 4, pp. 27-31, September 2016.
4. P. Eugster, V. Sundaram and X. Zhang, "Debugging the Internet of Things: The Case of Wireless Sensor Networks", IEEE Software, vol. 32, no. 1, pp. 38-49, 2015. Available: 10.1109/ms.2014.132.
5. A. S. Naik & R. Murugan, "An Energy Efficient Scheme for Secure Data Aggregation in Cluster Based Wireless Sensor Network", International Journal of Applied Engineering Research, vol. 13, no. 1, pp. 107-112, 2018.
6. M. Panda, "Security in Wireless Sensor Networks using Cryptographic Techniques", American Journal of Engineering Research (AJER), vol. 03, no. 1, pp. 50-56, 2014.
7. T. Ontagodi and G.R.D. Lakshmi, "Selecting key management schemes for WSN applications", International Journal of Recent Technology and Engineering (IJRTE), vol. 6, no. 4, pp. 14-17, September 2017.
8. F. Karray, M. Jmal, A. Garcia-Ortiz, M. Abid and A. Obeid, "A comprehensive survey on wireless sensor node hardware platforms", Computer Networks, vol. 144, pp. 89-110, 2018.
9. S. Md Zin, N. Badrul Anuar, M. Mat Kiah and I. Ahmedy, "Survey of secure multipath routing protocols for WSNs", Journal of Network and Computer Applications, vol. 55, pp. 123-153, 2015.
10. Y. Zhou, Y. Fang and Y. Zhang, "Securing wireless sensor networks: a survey", IEEE Communications Surveys & Tutorials, vol. 10, no. 3, pp. 6-28, 2008. Available: 10.1109/comst.2008.4625802.
11. A. Praveena, "Achieving data security in wireless sensor networks using ultra encryption standard version — IV algorithm", 2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT), 2017. Available: 10.1109/igeht.2017.8094068.
12. F. Zhang, R. Dojen and T. Coffey, "Comparative performance and energy consumption analysis of different AES implementations on a wireless sensor network node", International Journal of Sensor Networks, vol. 10, no. 4, p. 192, 2011. Available: 10.1504/ijsnet.2011.042767.
13. B. Mohd, T. Hayajneh and A. 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. Available: 10.1016/j.jnca.2015.09.001.
14. A. Al- Mamun, S. S. M. Rahman, T. Ahmed Shaon and M. Hossain, "Security Analysis of AES and Enhancing its Security by Modifying S-Box with an Additional Byte", International journal of Computer Networks & Communications, vol. 9, no. 2, pp. 69-88, 2017.
15. R. Riyaldhi, Rojali and A. Kurniawan, "Improvement of Advanced Encryption Standard Algorithm With Shift Row and S.Box Modification Mapping in Mix Column", Procedia Computer Science, vol. 116, pp. 401-407, 2017. Available: 10.1016/j.procs.2017.10.079.
16. A. S. Alkalbani, T. Mantoro, and A. O. M. Tap, "Comparison between RSA hardware and software implementation for WSNs security schemes," Proceeding of the 3rd International Conference on Information and Communication Technology for the Moslem World (ICT4M) 2010, 2010.
17. S. A. Verhade and N. N. Kasat, "A Review on Implementation of AES Algorithm Using FPGA and Its Performance Analysis," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 3, no. 1, pp. 404–408, 2015.
18. P. Wang, Y. Zhang, and J. Yang, "Research and Design of AES Security Processor Model Based on FPGA," Procedia Computer Science, vol. 131, pp. 249–254, 2018. Available: doi: 10.1016/j.procs.2018.04.210.
19. M. Cazorla, K. Marquet, and M. Minier, "Survey and Benchmark of Lightweight Block Ciphers for Wireless Sensor Networks," Proceedings of the 10th International Conference on Security and Cryptography, 2013.

20. 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. doi:10.1016/j.jnca.2015.09.001.
21. C. Panait and D. Dragomir, “Measuring the performance and energy consumption of AES in wireless sensor networks,” Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, 2015.: doi:10.15439/2015f322
22. I.C. Guzman, R.D. Nieto and A. Bernal, “FPGA implementation of the AES-128 algorithm in non-feedback modes of operation,” DYNA 83, vol. 198 pp. 37-43, 2016.
23. S.K. Reddy, R. Sakthivel and P. Praneeth. “VLSI implementation of AES crypto processor for high throughput.” International Journal of Advanced Engineering Sciences and Technology, vol. 6, no. 1, pp. 22-26.
24. Y. Wang and Y. Ha, “FPGA-Based 40.9-Gbits/s Masked AES with Area Optimization for Storage Area Network,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 60, no. 1, pp. 36–40, 2013.
25. H. Zodpe and A. Sapkal, “An efficient AES implementation using FPGA with enhanced security features,” Journal of King Saud University - Engineering Sciences, 2018.
26. F. V. Wenceslao Jr., “Performance Efficiency of Modified AES Algorithm Using Multiple S-Boxes.” International Journal of New Computer Architectures and their Applications, vol.5. no.1, pp.1-9, 2015.
27. A. R. Chowdhury, J. Mahmud, A. R. M. Kamal, and M. A. Hamid, “MAES: Modified advanced encryption standard for resource constraint environments,” 2018 IEEE Sensors Applications Symposium (SAS), 2018.
28. S. Chakraborty, A. Nachrani, A. Dasgupta, and P. Gajkumar, “Lightweight Security Protocol for WiSense based Wireless Sensor Network,” International Journal of Computer Applications, vol. 145, no. 3, pp. 6–10, 2016.
29. W. Stallings, “Cryptography and network security”, 4th ed. Upper Saddle River, N.J.: Pearson Prentice Hall, 2006, pp. 135-136.
30. Q. Liu, Y. Yuan, and Z. Xu, “High throughput and secure advanced encryption standard on field programmable gate array with fine pipelining and enhanced key expansion,” IET Computers & Digital Techniques, vol. 9, no. 3, pp. 175–184, 2015.
31. S. Hesham, M. A. A. E. Ghany, and K. Hofmann, “High throughput architecture for the Advanced Encryption Standard Algorithm,” 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, 2014.

AUTHORS PROFILE

Herman B. Acla earned his bachelor’s degree in computer engineering at Western Institute of Technology, Iloilo City, Philippines in 1999 and finished his Master of Engineering with specialization in Computer Engineering from the same institution on year 2009. He is an Assistant Professor IV at Northern Iloilo Polytechnic State College, Estancia, Iloilo. He is currently pursuing his Doctor of Engineering with specialization in Computer Engineering at Technological Institute of the Philippines, Quezon City, Philippines. His research interests includes, IT security, wireless sensor network, IoT, image processing and machine learning.

Bobby D. Gerardo a Professor VI of the College of ICT, is currently the Vice President for Administration and Finance of West Visayas State University, Iloilo City, Philippines. His dissertation is on Discovering Driving Patterns using Rule-based intelligent Data Mining Agent (RiDAMA) in Distributed Insurance Telematic Systems. He has published more than 100 research papers in national and international journals and conferences. He is a referee to international conferences and journals such as in IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Knowledge and Data Engineering, Elsevier Journal on Telematics, Future Generation Computer Systems and on Bioinformatics. His research fields are in the area of distributed systems telematics systems, data mining, deep learning, machine learning and IT security.