2019 6th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)

Security Analysis of Lightweight Encryption based on Advanced Encryption Standard for Wireless Sensor Networks

Herman B. Acla
Graduate Programs
TechnologicalInstitute of the Philippines
Quezon City, Philippines
hermanacla@gmail.com

Bobby D. Gerardo
Institute of Information and Communications Technology
West Visayas State University
Iloilo City, Philippines
bobby.gerardo@gmail

***Abstract*— Data security in Wireless Sensor Networks (WSNs) is very essential. However, because of the design of WSN devices with constrained resources; it is a challenge to develop and implement an encryption algorithm that is secure but will not consume much of the devices limited resources. This paper, presents a lightweight encryption based from Advanced Encryption Standard (AES) – LAES. In the proposed cipher, the MixColumns function of original AES is replaced by a 128-bit permutation to lessen the computational complexity of the algorithm and in order to reduce the resource utilization of the WSN node. During synthesis, results have shown that LAES have lesser device utilization as compared to recent implementation of traditional AES. The security of both LAES and AES were tested using the same sets of data to obtain the avalanche effect. Results have shown that on Plaintext bit flip, LAES obtained an average of 53.0469% which is 4.2969% higher than AES. On key bit flip, LAES obtained an average of 50.9375% which 3.4375% higher than AES. This shows that replacing MixColumns function with 128-bit permutation lessens the resource utilization in the hardware implementation of the encryption algorithm without compromising the security.**

Keywords*— *AES Cipher, Avalanche Effect, FPGA, Permutation, Wireless Sensor Network

I. INTRODUCTION

Wireless Sensor Network (WSN) is a network comprised of autonomous devices called motes or nodes equipped with sensors to monitor physical or environmental conditions communicating wirelessly with each other and with one or more base stations [1]. WSN draws its strength in its application to wide spectrum of discipline. According to [2] and [3], the application of WSN can be classified into two categories. First is monitoring applications which includes sensing environmental parameters, physiological parameters from patients, structural parameters, and others parameters from stationary subjects. Second is tracking which involves sensing parameters from moving subjects such as, animal behavior, vehicle tracking, traffic flow and humans. The nature of deployment of sensor nodes in open environments poses vulnerability to malicious attacks. In a mission critical application such as power installation and military, high-speed communication and security is essential. However, designing a security algorithm and implementing it to a resource constrained sensor nodes is challenging. Thus, the efficient utilization of these limited resources is critical [4].

Significant amount of security algorithm were proposed, examined and applied and each techniques has its strength and weaknesses [5]. However, Advanced Encryption Standard (AES) is the preferred cipher on embedded and WSN devices due to its efficiency in power and in ensuring the security of electronic information. It is a widely adopted strong cipher that can resist most of the security attacks [6], [7], and [8]. The work of [5], delved into the performance of Data Encryption Standard (DES), 3DES, AES, Rivest–Shamir–Adleman (RSA) and Blowfish. Result have shown that AES scored highest in terms of avalanche effect and recommends that AES should be used where confidentiality and integrity is critical in the operation in the field of its applications. Similarly in [9], the authors investigated three popular symmetric encryption algorithms namely RC5, AES and Skipjack; the result of their experiment revealed that AES performed better in terms of energy efficiency and security. The paper of [10] where they implemented AES algorithm in field-programmable gate array (FPGA), concluded that AES is dependable and practical in resisting password attacks. Moreover, in the works of [11], [12], and [13] demonstrated the efficiency of AES implementation in FPGA.

AES encryption is a block cipher that is composed of four core functions repeatedly performed in several rounds depending on the length of the encryption key used. To implement AES in WSN, a significant amount of computational operation has to be performed and it consumes considerable amount to the limited resources of the sensor node. According to [14] and [15], among the core functions of AES algorithm, MixColumns is the most complicated process and creates a bottleneck to the performance of the cipher. MixColumns is a type of diffusion operation that is critical in making the AES cipher secured. An assumption is considered that by replacing MixColumns with another diffusion function with lesser computational operation, will result to the improvement of the resource utilization of the cipher without compromising its security.

This paper will present an enhancement to the existing AES algorithm by introducing a bitwise permutation in place of MixColumns function and develop a lightweight version of AES. Device utilization of the proposed algorithm will be presented. Further, it also aims to determine its security in comparison to the original AES.

978-1-7281-4082-7 /19/$31.00 ©2019 IEEE

II. AES ALGORITHM

AES is a block cipher that processes data in 128-bits block in repetition. The number of repetition depends on the key length used. The key length can be specified as 128, 192, and 256 bits and the number of rounds with respect to key length is 10, 12, and 14 respectively [16]. AES is a symmetric encryption algorithm wherein the same key is used in the encryption and decryption process. Three variants of the cipher namely, AES-128, AES-192 and AES-256 depicts the key length used.

The overall structure of AES-128 cipher is shown in Fig. 1. The plaintext or the 128-bit block input of the cipher is expressed as a two dimensional 4 X 4 square matrix array of bytes called the State. Similarly, the key is expressed as square matrix of bytes and the ordering of bytes in the square matrix for both the plaintext and the key is sequenced by column. There are four phases used in the cipher: SubBytes, ShiftRows, MixColumns, and AddRoundKey. The cipher begins with AddRoundKey phase followed by nine rounds of all four phases and the last round without the MixColumn phase.

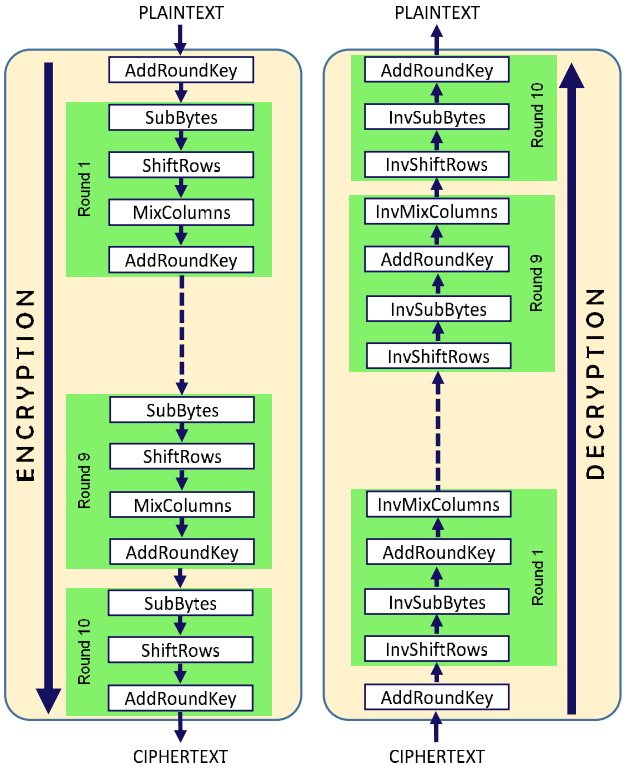

Fig. 1. AES cipher function.

A. SubBytes Phase

SubBytes, or substitute byte transformation is a byte by byte substitution of the block using a substitution box known as S-Box. S-Box contains a permutation of all possible 256 8-bit values depicted as 16 X 16 matrix of byte values. SubBytes operation is shown in Fig.2. Each individual byte of State is mapped into a new byte in the following way: The leftmost 4 bits of the byte are used as a row value and the rightmost 4 bits are used as a column value. These row and column values serve as indexes into the S-box to select a unique 8-bit output value.

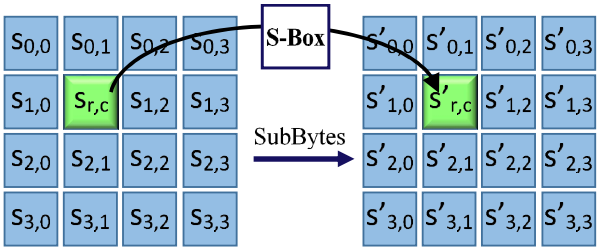

Fig. 2. Substitute byte transformation.

B. ShiftRows Phase

ShiftRows or shift row transformation is a simple permutation. The structure of ShiftRow operation is shown in Fig.3 in which a byte circular left shift is performed to the last three rows of the State matrix. The first row of State is not altered. For the second row, one byte circular shift to the left of the matrix is performed. For the third row, two bytes circular shift to the left of the matrix is performed. For the fourth row, three bytes circular shift to the left of the matrix is performed.

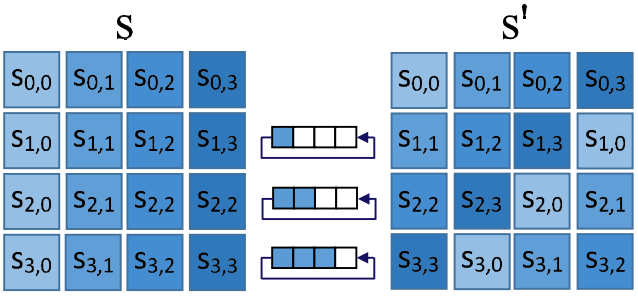

Fig. 3. Shift row transformation.

C. MixColumns Phase

MixColumns or mix columns transformation is shown in Fig.4. It is a linear diffusion process that operates on each of the state matrix column individually. The transformation is 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).

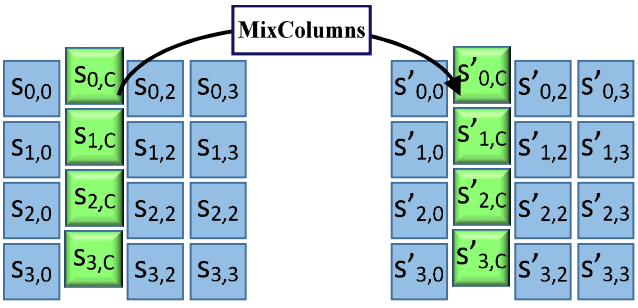

Fig. 4. Mix columns transformation

D. AddRoundKey Phase

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.

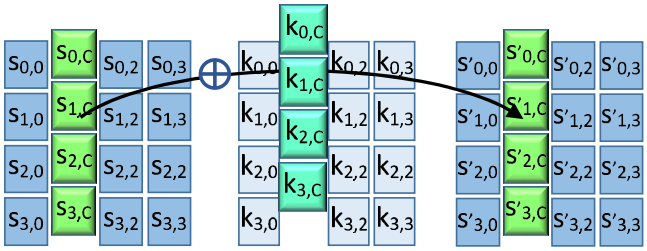

Fig. 5. Add round key transformation.

III. Proposed Enhanced AES Algorithm

Replacing MixColumns of AES with a 128-bit Permutation resulted in an increase in resource utilization, throughput and performance efficiency [17]. The overall structure of the lightweight AES (LAES) algorithm is shown in Fig. 6. The same with AES-128, LAES uses a 128-bit block size input and 128-bit key length. Encryption process of the LAES is divided into three stages. First is initial round stage, where process begins with the plaintext converted into a 4 X 4 matrix array known as State and bitwise XORed to the 128 bits of round key. Second stage is nine rounds of SubBytes, ShiftRows, Permutation, and AddroundKey transformations are performed to State. Third and final stage is the last round where SubBytes, ShiftRows, and AddroundKey transformations are performed to State to produce the Ciphertext.

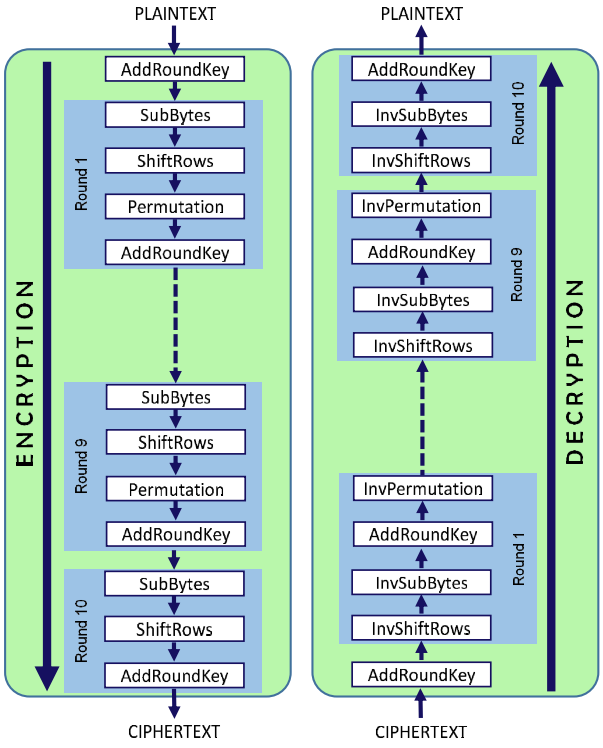

Fig. 6. Proposed lightweight AES cipher structure.

The motivation for the substituting MixColumns with Permutation process is that MixColumns involves complex and numerous calculation thus, directly affects the resource utilization and performance of the algorithm during encryption and decryption.

The Permutation transformation is a diffusion operation wherein each of the 128 bits input is spread over the 128 bits output. Input bits, are jumbled based on a 128 bit permutation table. After the permutation stage, bit number 1 is at bit number 122, bit number 2 is at bit number 114, and so on; until bit 128 is at number 7. Shown in Fig 7 is the description of Permutation transformation. To look at the design in the electrical circuit point of view, it's like putting a wire from point 1 to point 122, wire from point 2 to point 114 until point128 to point 7. With this, a lightweight version of AES is realized.

The decryption process of LAES involves three stages. First is initial round where Ciphertext as input is XORed to the initial round key which is the same to the round key used in round 10 of the Encryption process. Second is nine rounds of InvShiftRows, InvSubBytes, AddroundKey and InvPermutation transformations. Description of the InvPermutation transformation is shown in Fig. 8. Third is the last round of InvShiftRows, InvSubBytes and AddRoundKey transformations to produce the Plaintext.

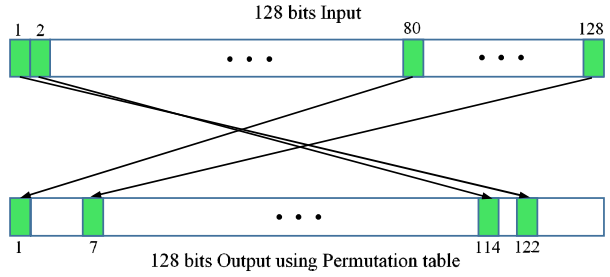

Fig. 7. Bitwise Permutation

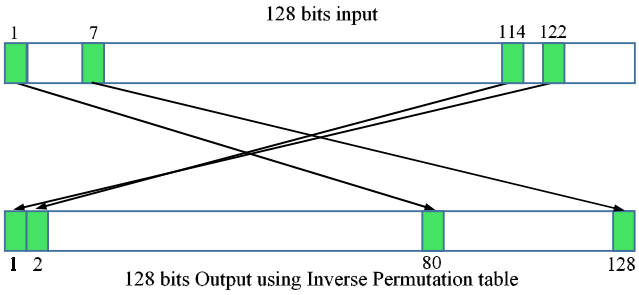

Fig. 8. Inverse Bitwise Permutation

IV. Implementation Results and Analysis

Very High Speed Integrated Circuit Hardware Description Language (VHDL) programming was used in developing the LAES algorithm; a hardware description language (HDL) used to program FPGAs. Virtex-7 XC7VX690T was used for hardware implementation of the proposed Enhanced AES algorithm. Xilinx ISE Web Pack 14.7 was utilized in the synthesis of the algorithm and ISE Simulator (ISim) was used in the simulation.

A. Device Utilization

Replacing MixColumns with Permutation made the AES algorithm lighter in terms of device utilization as shown in

the following table. Compared to previous implementation of [18], LAES has lesser slice registers and slice LUTs.

TABLE I. COMPARISON OF DEVICE UTILIZATION

Parameters	LAES	[18]
No. of Slice Registers	858 out of 866400	3760 out of 866400
No. of Slice LUTs	2715 out of 433200	10773 out of 433200
No. of Bonded IOBs	79 out of 600	385 out of 850
No. of LUT Flip Flop Pairs	2729	13068

B. Time Security

It depicts the strength of cryptographic algorithm against brute force attack with different key size and time it takes to effectively mount a brute force attack. Brute force attack is a trial and error method through exhaustive effort of decoding encrypted data. Given that LAES uses a 128-bit key, the maximum key combination is 2^{128} or 3.403 x 10^{38}.

To date, the fastest supercomputer in the world is Summit by IBM Systems. The recorded speed is 148.6 x 10^{15} floating points operation per second or 148.6 PetaFLOPS [19]. Assuming using this supercomputer to crack LAES in brute force attack, it will take 7.2563 x 10^{13} years for it to break. Making it impossible for a human or even a generation to crack the encryption by testing all the possible combination. Since LAES and AES-128 uses the same key length of 128, therefore they have equal strength against brute force attack.

C. Avalanche Effect

A desired property of any cryptographic algorithm, hash function and cipher block encryption is the avalanche effect. It implies that in any encryption algorithm, a slight change in either the plaintext or the key will result to a significant change in the ciphertext [16]. Specifically, if one bit is changed in the plaintext or the key, it should produce a change in many bits of the ciphertext. If the change were small, the cryptologists can guess the plaintext by analyzing the ciphertext and eventually making it possible for them to break the algorithm. Based on [20] avalanche effect, can be calculated using the equation:

$$Avalanche\ Effect = \frac{NBC}{TNB} * 100\% \quad (1)$$

Where NBC is the number of bits changed in the Ciphertext after the bit flip and TNB is the total number of bits in the Ciphertext.

To determine the avalanche effect on LAES, two sets of experiment were conducted. First, a single bit change is made on the plaintext while the key remains the same. In Test 1, the plaintexts are "123456789ABCDEF0123456789ABCDEF0" and "123456789ABCDEF0123456789ABCDEF1". Using key "11111111111111111111111111111110", the two plaintexts were encrypted separately. The resulting Ciphertexts were "86B3CC7080993ADF2466BDE8BA31E10D" and on single bit change is "A5CA414DB1780B84104C4C1BF6CE3542". Using equation (1), the avalanche effect is 53.90625%.

The same set of plaintexts and key were used and encrypted in AES to determine the avalanche effect. In Test 1, "123456789ABCDEF0123456789ABCDEF0" and "123456789ABCDEF0123456789ABCDEF1" are the plaintexts. Using "11111111111111111111111111111110", as key, the Ciphertexts "171434671D73293B813735A3F0729FBF" and "136ED3E12AAE2B10C0816C286BA91095" was recorded respectively. The avalanche effect was computed using equation (1), and resulted to 50.7813%.

Ten pairs of plaintexts and keys were used during the testing of avalanche effect when the bit change is made on the plaintext both on LAES and AES. The summary of the tests is shown in the following table. The average avalanche effect of LAES on plaintext bit flip is 53.0469% and 48.7500% on AES. Based on this result, LAES is 4.2969% better than AES in terms of avalanche effect.

TABLE II. AVALANCHE EFFECT OF LAES ON PLAINTEXT BIT FLIP

Test	Plaintext	Ciphertext	Avalanche Effect (%)
1	123456789ABCDEF0 123456789ABCDEF0	86B3CC7080993ADF2 466BDE8BA31E10D	53.9063
	123456789ABCDEF0 123456789ABCDEF1	A5CA414DB1780B841 04C4C1BF6CE3542	
2	112233445566778899 AABBCCDDEEFF00	47588696C34ECBA984 C793BFCA558788	59.3750
	112233445566778899 AABBCCDDEEFF01	4024CD6933A801593D 2A6901C8C73035	
3	1EE823570972BB0F3 0D05938C132D612	B79A02704036BF5F4D 4FEB1D178B053C	51.5625
	1EE823570972BB0F3 0D05938C132D613	5475E12FA43536DFA4 18975616C2CB92	
4	E1172357097244F030 D059373ECD2944	24D3BB661FDE9C9C5 6A824A5734A3637	53.9063
	E1172357097244F030 D059373ECD2945	CD74CE07D24357FC1 C2F03F605619EE8	
5	001122334455667788 99AABBCCDDEEFF	44ABD3DDCB8142B1 85F9F332E127449F	53.1250
	001122334455667788 99AABBCCDDEEFE	8C73E4A7CE20941C5 6B3AA0757B0C300	
6	5452555354204E4F20 4F4E4521585858	CAAA627F51695A0B1 99F560C6B269586	51.5625
	5452555354204E4F20 4F4E4521585859	3BFFB9C5C9E0A30B9 C60D08D804E30BB	
7	4A454E53454E53454 154484F41434C41	D85DEF2C5A56F99C ABE039203DC86CF6	48.4375
	4A454E53454E53454 154484F41434C40	924B21CFC20744CF0 B1894A6EB40CEC1	
8	41636C612C4A616B6 520526F756B6500	E87185F1CF8A7C35D 055C923D66CD679	50.0000
	41636C612C4A616B6 520526F756B6501	6713970A04102EA7F3 B8DB3BBA8A00A5	
9	41434C414A494E445 24F414C57594E4E	86A3B35DAEBA02B8 B7EC12BA3C599B4D	53.9063
	41434C414A494E445 24F414C57594E4D	026D148DE7AEA1390 DFF566A166092D2	
10	4D59204D455353414 74520495320494E	38B9297A7EBA8E797 2BEBDA2EF94CCCA	54.6875
	4D59204D455353414 74520495320494D	C685C73FB9F5CD0A9 1042A327B23748E	
		Average Avalanche Effect	53.0469

TABLE III. AVALANCHE EFFECT OF AES ON PLAINTEXT BIT FLIP

Test	Plaintext	Ciphertext	Avalanche Effect (%)
1	123456789ABCDEF0 123456789ABCDEF0	171434671D73293B813 735A3F0729FBF	50.7813
	123456789ABCDEF0 123456789ABCDEF1	136ED3E12AAE2B10C 0816C286BA91095	
2	112233445566778899 AABBCCDDEEFF00	399DF9D05F2F2DBE8 AD152FA7524A119	46.0938
	112233445566778899 AABBCCDDEEFF01	193FA56F0A3D51CF0 004375F65920203	
3	1EE823570972BB0F3 0D05938C132D612	0E57A14F014A72FFF3 C15797C5E1DE3E	41.4063
	1EE823570972BB0F3 0D05938C132D613	0C4597074AF2BBAB7 F9912E77AF65ED7	
4	E1172357097244F030 D059373ECD2944	F2F4AE51AA32B6D9 AD3FAE4E1CC6BE0D	48.4375
	E1172357097244F030 D059373ECD2945	783F4BE1181F60166E 602F0E0F1A0608	
5	00112233445566778 899AABBCCDDEEFF	A73C1D277CEC94F80 96BCE6C6D68DD86	48.4375
	00112233445566778 899AABBCCDDEEFE	9B82515830C88086DE 3BC07DDB6FEBB5	
6	5452555354204E4F20 4F4E4521585858	A94E74186D51A8A82 E09812F44DD9229	50.0000
	5452555354204E4F20 4F4E4521585859	BFD480B9FBF519A70 3C1F6FEB029A863	
7	4A454E53454E53454 154484F41434C41	A3C9F6864A505DA5F FB80545C85AB67F	53.1250
	4A454E53454E53454 154484F41434C40	4F93E0933FAA8347EE 678625278080EF	
8	41636C612C4A616B6 520526F756B6500	035059139A68B661D2 D8628D2604AF09	53.1250
	41636C612C4A616B6 520526F756B6501	EF212A45DB9483B0D FB739D1B2615CA8	
9	41434C414A494E445 24F414C57594E4E	4967501DA39E121D90 E73096D2327448	49.2188
	41434C414A494E445 24F414C57594E4D	55DE8695D1106B7689 2C167882C2976A	
10	4D59204D455353414 74520495320494E	233201384A166438AD 530B688E312801	46.8750
	4D59204D455353414 74520495320494D	1B74CF3B931B8BAA8 727AAA195EBAE84	
		Average Avalanche Effect	48.7500

In the second set of experiment, the same set of Plaintext with single bit change on the key were used. In Test 1, "123456789ABCDEF0123456789ABCDEF0" was used as plaintext while "11111111111111111111111111111110" and "11111111111111111111111111111111" was assigned as keys. The resulting Ciphertexts after the encryption in LAES are "86B3CC7080993ADF2466BDE8BA31E10D" and "5E04AD87865C0F309EBC3E43D0555F71" respectively. Using equation (1), the computed avalanche effect is 57.03125%. The same set of Plaintext and keys were implemented to AES and resulted to "171434671D73293B813735A3F0729FBF" and "1CEDA86D5B5C677163D6407EF417C533" as Ciphertexts respectively with an avalanche effect of 47.65625%.

Ten sets of data were used on LAES and AES where the same plaintexts were used and a bit flip is done on the key. The average avalanche effect on LAES on key bit flip is 50.9375% and 47.50% on AES. Based on this result, LAES is 3.4375% better than AES in terms of avalanche effect.

Shown in Fig. 9 is the comparison between the avalanche effect of LAES and AES when the bit change is in the plaintext while the comparison between the avalanche effects of AES and LAES when the bit change is in the key is shown in Fig.10.

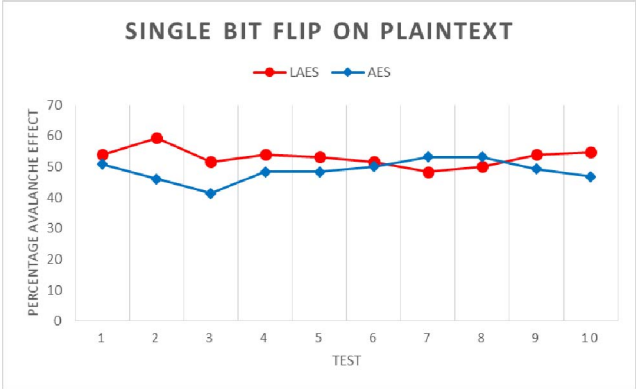

Fig. 9. Avalanche effect on plaintext bit change.

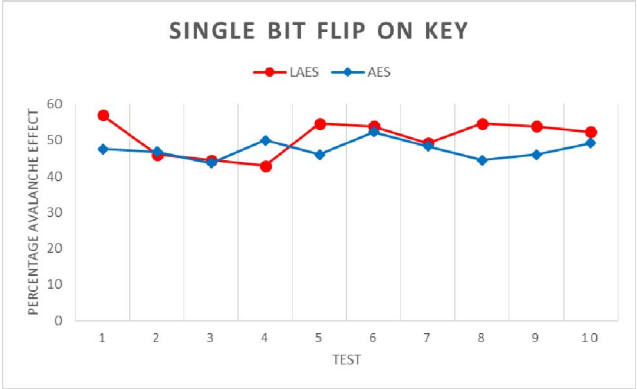

Fig. 10. Avalanche effect on key bit change.

V. IMPLEMENTATION RESULTS AND ANALYSIS

Encryption is generally used to secure data. In this paper, LAES, an enhanced lightweight cipher based on AES designed for WSN was presented. The MixColumns of AES was replaced by a bitwise permutation to lessen the computational complexity of the algorithm. The time security was computed and using the latest and fastest supercomputer, it would take 7.2563x1013 years to break the LAES using brute force attack.

Further, two sets of experiments were conducted to obtain the avalanche effect of the proposed LAES. First, a single bit change on plaintext with the same key. Second, the same plaintext were used on separate keys with single bit change. The same ten sets of data were implemented on LAES and AES. On Plaintext bit flip, LAES obtained an average of 53.0469% while AES obtained an average of 48.75% showing that LAES achieved 4.2969% higher than AES. On key bit flip, LAES obtained an average of 50.9375% while AES obtained an average of 47.50% showing that LAES achieved 3.4375% higher than AES.

Based from these results, we conclude that replacing MixColumns function of AES with 128-bit permutation improved the security of the cipher in terms of avalanche effect.

ACKNOWLEDGMENT

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

REFERENCES

[1] Y. Gao, H. Ao, Z. Feng, W. Zhou, S. Hu, and W. Tang, "Mobile network security and privacy in WSN, *Procedia Computer Science*, 129, pp. 324-330, 2018.

[2] S. Zhang, and H. Zhang, "A review of wireless sensor networks and its applications," *2012 IEEE International Conference on Automation and Logistics*. 2012.

[3] M. Elhoseny, A. Tharwat, X. Yuan, and A. E. Hassanien, "Optimizing K-coverage of mobile WSNs," *Expert Systems with Applications,* 92, pp. 142–153, 2018.

[4] N. Bandirmali, and I. Erturk, "WSNSec: A scalable data link layer security protocol for WSNs," *Ad Hoc Networks,* vol. 10, no. 1, pp. 37–45, Jan. 2012.

[5] P. Patil, P. Narayankar, D.G. Narayan, and S.M. Meena, "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish," *Procedia Computer Science,* vol. 78, pp. 617–624, 2016.

[6] C. Manifavas, G. Hatzivasilis, K. Fysarakis, and Y. Papaefstathiou, "A survey of lightweight stream ciphers for embedded systems," *Security And Communication Networks,* vol. 9, no. 10, pp. 1226–1246 , 2015.

[7] T. Abdelmoghni, O. Z. Mohamed, B. Billel, M. Mohamed, and L. Sidahmed, "Implementation of AES coprocessor for wireless sensor networks," *2018 International Conference on Applied Smart Systems (ICASS)*, 2018.

[8] N. Shaji, and P.L. Bonifus, "Design of AES architecture with area and speed tradeoff," *Procedia Technology*, vol. 24, pp. 1135–1140, 2016.

[9] S. Sathyadevan, S. Prabhakaranl, and K. Bipin, "A survey of security protocols in WSN and overhead evaluation," Advances in Intelligent Systems and Computing Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014, pp. 729–738, 2015.

[10] S. S. Chorage and V. A. Somwanshi, "Fault resistant encryption system using high speed AES algorithm on FPGA," *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, pp 466–470, 2017.

[11] A. P. A Naidu and P. K. Joshi, "FPGA implementation of fully pipelined Advanced Encryption Standard," *2015 International Conference on Communications and Signal Processing (ICCSP),* pp. 649–653, 2015.

[12] S. M. U. Talha, M. Asif, H. Hussain, A. Asghar, and H. Ameen, "Efficient advance encryption standard (AES) implementation on FPGA using Xilinx system generator," *2016 6th International Conference on Intelligent and Advanced Systems (ICIAS),* 2016.

[13] S. U. Jonwal and P.P Shingare, "Advanced Encryption Standard (AES) implementation on FPGA with hardware in loop," *2017 International Conference on Trends in Electronics and Informatics (ICEI),* pp. 64–67, 2017.

[14] C. Lu, Y. Kao, H. Chiang and C. Yang, "Fast implementation of AES cryptographic algorithms in smart cards," *IEEE 37th Annual 2003 International Carnahan Conference on Security Technology, 2003 Proceedings,* 2004.

[15] Ratnadewi, R. Adhie, Y. Hutama, J. Christian, and D. Wijaya, "Implementation and performance analysis of AES-128 cryptography method in an NFC-based communication system," *World Transactions on Engineering and Technology Education.* vol. 15, pp. 178-183, 2017.

[16] W. Stallings. *Cryptography and Network Security - Principles and Practice,* 6th ed.**,** Upper Saddle River, New Jersey: Pearson Education Limited, 2014, pp. 130-155.

[17] H. Acla and B. Gerardo, "Performance evaluation of lightweight Advanced Encryption Standard hardware implementation, *International Journal of Recent Technology and Engineering (IJRTE),* vol. 8, no. 2, pp. 1810-1815, July, 2019.

[18] N. S. S. Srinivas and M. Akramuddin, "FPGA based hardware implementation of AES Rijndael algorithm for Encryption and Decryption," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 1770–1776, 2016.

[19] Y. Sverdlik. "The world's 10 fastest supercomputer – in pictures," *Data Center Knowledge,* 2019. [Online]. Available:https://www.datacenterknowledge.com/supercomputers/world-s-10-fastes-supercomputers-pictures. [Accessed: 03-Aug-2019].

[20] C. P. Dewangan and S. Agrawal, "A novel approach to improve avalanche effect of AES algorithm," *International Journal of Advanced Research in Computer Engineering & Technology,* Volume 1, Issue 8, October 2012.