The Scientific Temper (2023) Vol. 14 (1): 184-190

E-ISSN: 2231-6396, ISSN: 0976-8653

Doi: 10.58414/SCIENTIFICTEMPER.2023.14.1.22

https://scientifictemper.com/

RESEARCH ARTICLE

Enhanced AES-256 cipher round algorithm for IoT applications

Santhanalakshmi Mahalingam*, Lakshana Kathirvel and Shahitya G. Maheswari

Abstract

Objectives: Networks have become a significant mode of communication in recent years. As a result, internet security has become a critical requirement for secure information exchange. Cryptography is used to securely send passwords over large networks. Cryptographic algorithms are sequences of processes used to encipher and decipher messages in a cryptographic system. One of those is the Advanced Encryption Standard (AES), which is a standard for data encryption in hardware and software to hide sensitive and vital information. The main objective is to design an AES system with modifications by the addition of primitive operations which can withstand several attacks and is more efficient.

Method: AES works with three different key lengths: 128-bit keys, 192- bit keys, and 256-bit keys. The early rounds of AES have a poor diffusion rate. Better diffusion properties can be brought about by putting in additional operations in the cipher round and key generation algorithm of the conventional AES.

Findings: The diffusion characteristics of the conventional AES and the proposed methodology are compared using the avalanche effect. The proposed AES algorithm shows an increased avalanche effect, which proves it to be more secure than the conventional AES. The proposed algorithm is executed on Vivado 2016.2 ISE Design Suite and the results are targeted on Zybo–Zynq Z-7010 AP SoC development board.

Novelty: In addition, this paper also proposes an improved AES algorithm that was accomplished by altering the sub-bytes operation. This change was made to make it more reliant on round keys. This algorithm was also extended to a higher key length of 256 bits which makes the algorithm less vulnerable to attacks.

Keywords: AES, Cryptography, Decryption, Enhanced security, Encryption.

Introduction

Cryptography is a set of techniques for storing and transmitting information while keeping it safe from intruders. Cryptography achieves this by converting data into an incomprehensible form (called ciphertext or code) (Gupta, 2020). Encryption converts the original message into a ciphertext and the encrypted data is converted back to its original message through decryption. The sender and receiver need secret key to perform encryption and decryption. This key should not be disclosed to third parties to perform secure communication.

Several applications of modern cryptographic theory in hardware and software are expected to be available in the near future (Wood, 2012). Hardware implementation, as contrasted to software, offers better physical security and speed. One such means, the X86 architecture, as a CISC (Complex Instruction Set Computer) Architecture, implements significant components of the AES algorithm. It can be used by the NSA (National Security Agency) for top-secret information. The architecture also supports hashing algorithms which can be used for password verification. A part of the processor is dedicated to encryption and decryption in hardware implementation. As a result, speed is greatly increased, and even if the operating system is compromised, the data is still secure. However, it is more problematic to solve if a hardware implementation is compromised in case of attacks.

The Advanced Encryption Standard (AES) is a symmetric type of encryption that the United States government has chosen to safeguard confidential information in both software and hardware. Most of the existing AES hardware

Department of Electronics and Communication Engineering, PSG College of Technology (Autonomous), Coimbatore, Tamil Nadu, India

***Corresponding Author:** Santhanalakshmi, M., Department of Electronics and Communication Engineering, PSG College of Technology (Autonomous), Coimbatore, Tamil Nadu, India

, E-Mail: ms.ece@psgtech.ac.in

How to cite this article: Santhanalakshmi, M., Ms. Lakshana, K., Shahitya, G. M. (2023). Enhanced AES-256 cipher round algorithm for IoT applications. The Scientific Temper, 14(1):184-190

Doi: 10.58414/SCIENTIFICTEMPER.2023.14.1.22

Source of support: Nil

Conflict of interest: None.

Received: 04/01/2023 **Accepted:** 05/02/2023 **Published** : 25/03/2023

design implementations could be improved in terms of security (Soliman *et al.,* 2016). The current study mainly focuses on the hardware implementation of existing AES systems with modifications for better security and efficiency.

Field Programmable Gate Arrays (FPGA) are more flexible, less complex and hence provide more efficiency. AES design that is implemented on Artix-7 FPGA gives a clear idea of the resources needed for the hardware implementation (Kumar, Ramkumar and Kaur, 2020). It can be noted that AES can be built with hardware as well as implemented in software (Borkar, Kshirsagar and Vyawahare, 2011). But FPGAs are efficient and customizable solutions for AES implementation. Software is used for optimizing the VHDL code, and an iterative design approach is used to minimize hardware consumption. A high-speed AES algorithm is implemented on FPGA by improving the computing speed of the system and using pipelining and parallel processing methods (Wang, Chen and Xu, 2012). The introduction of mathematical principles and logic structure of AES algorithm is very helpful for understanding the rounds in AES, so as to modify it to improve security. By reducing the number of slices for AES design in VHDL, area is optimized to a great extent in , thereby increasing the efficiency of the algorithm (Deshpande et al., 2014). For optimization, all the components of the AES algorithm are examined, which gives a clear idea for AES optimization, paving the way to achieve good performance.

AES is optimized and the DESI (Data Encryption Standard in IoT) in the Internet of Things environment is elucidated by Su, Zhang and Li (2019). To test the efficiency of the designed encryption algorithm, encryption and decryption tests are performed on data of different sizes (Setetemela *et al.,* 2019). Thus, for each algorithm, the average value is obtained by multiple tests. This paper adopts this idea to compare the proposed algorithm with the conventional AES algorithm. By using the mixing of columns and Inverse mixing of column operation, a high-performance and area-efficient approach for AES is proposed by Parikh and Narkhede (2016). With this, different methods for implementing s-box, mixing of columns and inverse mixing of columns are inferred, which were helpful for the modification of conventional AES in this paper. To drastically increase the security of encryption, the round function of AES algorithm is modified by Talirongan, Sison & Medina (2018). The butterfly effect is used to bring about novelty in altering the round function of AES, thereby improving the security of both the encryption and decryption processes. Three security level measurements are utilized for the same: the degree of diffusion, confusion, and integrity check. This approach was useful in modifying the round function of conventional AES and measuring diffusion characteristics of the proposed AES in this paper. Each round operation of AES is piped into 4 stages, where the decryption module reuses some circuits of the encryption module (Yuan et al., 2018). This improves the performance of the algorithm in terms of area and throughput. Thus, a basic idea for reusing some circuits for better performance is inferred through this study.

Table 1: Number of rounds – AES algorithm

Type	*Key Length (bit)*	*Block Size*	*Number of Round*
AES-128	128	128 bit	10
AES-192	192	192 bit	12
AES-256	256	256 bit	14

After a thorough analysis of the above papers by De Los Reyes, Sison and Medina (2018), as the first rounds of conventional AES have a low diffusion rate, incorporating elementary operations like XOR and modulo arithmetic in each round is thought to be an efficient strategy for hardware implementation of AES. It also proves for better security due to an increase in randomness, which can be proved using the avalanche effect.

Conventional AES

The conventional AES algorithm (also known as the Rijndael algorithm) is a symmetrical encryption algorithm that converts plain text in 128-bit blocks to cipher text employing keys of 128, 192, and 256 bits. The AES algorithm is a worldwide standard because it is considered secure. To generate cipher text, the AES algorithm employs a substitution-permutation (SP) network with many rounds, which is elaborated by Su, Zhang and Li (2019). The total number of rounds is being determined on key size. Table 1 shows the key length for different types of AES algorithm (Elmogy et al., 2019) and each round in the algorithm consists of four steps, which is shown in Figure 1.

ADDROUND KEY

The AddRoundKey function is the only part of the AES encryption process that directly works on the round key. The input to this particular round and the generated round key are XORed in this addround key operation.

SUB BYTES

Splitting the input into bytes and sending each through a

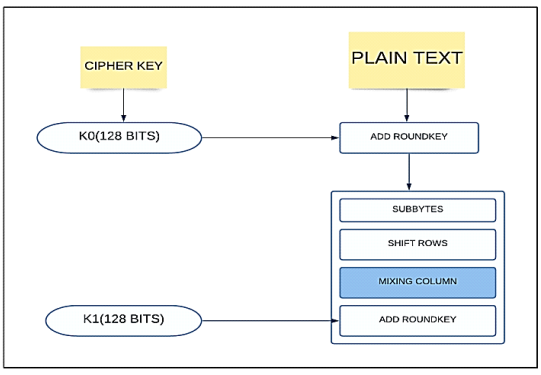

Figure 1: Round in AES Algorithm

Substitution Box or S-Box is the Sub-Bytes phase of AES. In contrast to DES, AES employs the same S-Box for all bytes. In Galois Field 2^8, the S-Box has all the possible permutations of 256 values. The AES S-Box is shown in Table 2.

SHIFTROWS

Every row in the cipher's state is being shifted through Shift Rows module. Each of the rows is being shifted to left, with the row number beginning at zero. The upper row has no shift, the subsequent row has one shift, and so on.

MIXCOLUMNS

The MixColumns segment, just like the ShiftRows stage of AES, delivers diffusion by mixing the input around. MixColumns, divides the matrix into columns. MixColumns, different ordinary matrix multiplication, implements matrix multiplication rendering to Galois Field 2^8.

AES 128 – Encryption and Decryption

The AES 128 algorithm is implemented by bringing about changes in: (1) the key generation algorithm where prior to creation of the subkeys, an extra substitution byte step and round constant addition are performed (De Los Reyes, Sison and Medina, 2018). This is done to prevent direct use of the key, (2) the cipher round algorithm, where more elementary operations were added, such as XOR and modulo addition at the early rounds of AES 128 to increase the rate of diffusion. The avalanche effect was used to compare the modified AES 128 to the conventional AES in order to quantify diffusion properties.

In this work, AES encryption described by De Los Reyes, Sison and Medina (2018) is extended to 256 bit key instead of the 128 bit key. Since the key length increases, security increases, which is proved by the avalanche effect. Also, the more the rounds, the more complex the encryption, which makes the algorithm practically impregnable by brute force assaults.

In decryption, all operations done in encryption, such as Shift Row, sub-bytes, Mix Columns, and modulo addition, will be reverted to inverse Shift Row, inverse sub-bytes, and inverse Mix Columns and modulo subtraction, all using their inverses.

Proposed Methodology

This work proposes an improved AES algorithm by modifying the sub-bytes operation to make it more reliant on round keys. This ensures that even minor changes in key are detected in the cipher.

4.1 Proposed AES 256 – Encryption

The conventional AES algorithm is modified by introducing changes in the sub-byte operation. This is achieved by the following steps: Four 8-bit keys XORk0, XORk1, XORk2, XORk3 are obtained using the 16 bytes round key. Every byte in the corresponding row is XORed in the key matrix of that round, as in Eq 4.1. Further, each XORki is xored to corresponding byte in the row of the state matrix prior to s-box substitution. The state matrix and the round key are represented as given below:

$$S = \begin{matrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{matrix}$$

Table 2: S Box

	0	1	2	3	4	5	6	7	8	9	0a	0b	0c	0d	0e	0f
0	63	7c	77	7b	f2	6b	6f	c5	30	1	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	CC	34	a5	e5	f1	71	d8	31	15
30	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
40	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	2	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	8
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

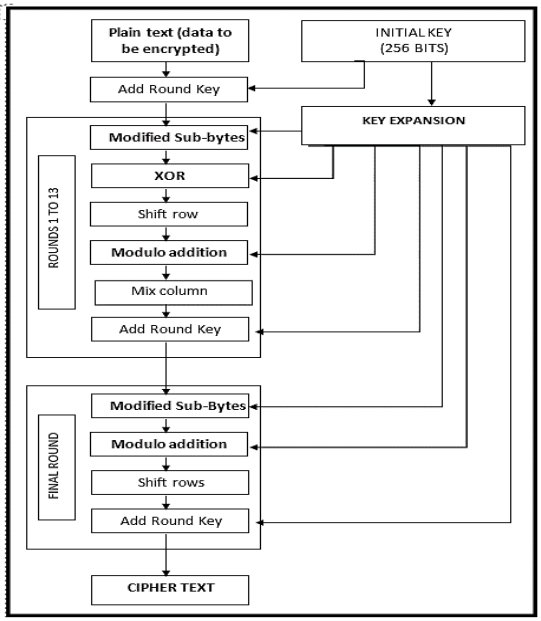

Figure 2: Proposed AES 256 algorithm – Encryption

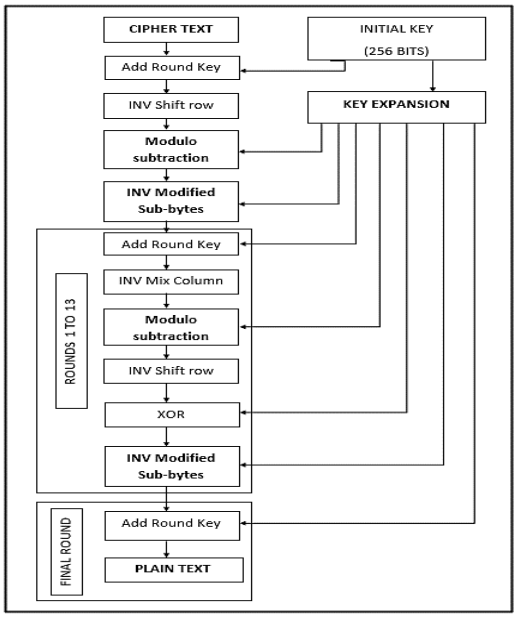

Figure 3: Proposed AES 256 Algorithm – Decryption

$$K = \begin{matrix} K_{0,0} & K_{0,1} & K_{0,2} & K_{0,3} \\ K_{1,0} & K_{1,1} & K_{1,2} & K_{1,3} \\ K_{2,0} & K_{2,1} & K_{2,2} & K_{2,3} \\ K_{3,0} & K_{3,1} & K_{3,2} & K_{3,3} \end{matrix}$$

The EXOR operation is as follows:

$$XORk_i = K_{i,0} \oplus K_{i,1} \oplus K_{i,2} \oplus K_{i,3}, \text{ where } i = 0 \text{ to } 3 \quad (4.1)$$

$$XORk_0 = K_{0,0} \oplus K_{0,1} \oplus K_{0,2} \oplus K_{0,3} \quad (4.2)$$

$$XORk_1 = K_{1,0} \oplus K_{1,1} \oplus K_{1,2} \oplus K_{1,3} \quad (4.3)$$

$$XORk_2 = K_{2,0} \oplus K_{2,1} \oplus K_{2,2} \oplus K_{2,3} \quad (4.4)$$

$$XORk_3 = K_{3,0} \oplus K_{3,1} \oplus K_{3,2} \oplus K_{3,3} \quad (4.5)$$

Eq. 4.6 gives the new state matrix S'

$$S'_{i,j} = S_{i,j} \oplus XORk_i \quad (4.6)$$

where j = 0 to 3, i = 0 to 3

The EXOR operation is as follows

$$S' = \begin{matrix} S_{0,0} \oplus K_0 & S_{0,1} \oplus K_0 & S_{0,2} \oplus K_0 & S_{0,3} \oplus K_0 \\ S_{1,0} \oplus K_1 & S_{1,1} \oplus K_1 & S_{1,2} \oplus K_1 & S_{1,3} \oplus K_1 \\ S_{2,0} \oplus K_2 & S_{2,1} \oplus K_2 & S_{2,2} \oplus K_2 & S_{2,3} \oplus K_2 \\ S_{3,0} \oplus K_3 & S_{3,1} \oplus K_3 & S_{3,2} \oplus K_3 & S_{3,3} \oplus K_3 \end{matrix}$$

Therefore, the final state matrix is given by

$$S' = \begin{matrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{matrix}$$

Using normal Sub-Bytes operation as in Eq. 4.7, the bytes are substituted in the s-box after obtaining S'

$$S'_{i,j} = SubstitutionBox\,[S'_{i,j}] \quad (4.7)$$

where j = 0 to 3 for every i = 0 to 3

The Proposed AES 256 algorithm – Encryption block is described in Figure 2.

Proposed AES Algorithm-Decryption

It is proven that the Sub-Bytes transformation is invertible. The inverse Sub-Bytes operation of the proposed algorithm is as follows:

$$S = \begin{matrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{matrix}$$

The original state matrix S is obtained by

$$S = \begin{matrix} S'_{0,0} \oplus K_0 & S'_{0,1} \oplus K_0 & S'_{0,2} \oplus K_0 & S'_{0,3} \oplus K_0 \\ S'_{1,0} \oplus K_1 & S'_{1,1} \oplus K_1 & S'_{1,2} \oplus K_1 & S'_{1,3} \oplus K_1 \\ S'_{2,0} \oplus K_2 & S'_{2,1} \oplus K_2 & S'_{2,2} \oplus K_2 & S'_{2,3} \oplus K_2 \\ S'_{3,0} \oplus K_3 & S'_{3,1} \oplus K_3 & S'_{3,2} \oplus K_3 & S'_{3,3} \oplus K_3 \end{matrix}$$

The Proposed AES 256 Algorithm – Decryption block is elaborated in Figure 3.

In the proposed AES 256, there is more randomness in each of the 14 rounds than the conventional AES. The increase in randomness makes it hard to predict the cipher text, making it difficult for the attackers to extract the data. Thus, the proposed AES 256 aims to have better efficiency and security than the conventional AES algorithm by withstanding several attacks.

Implementation and Validation

Simulation of Proposed AES-256

AES-256 encrypts and decrypts a block of messages with a 256-bit key length. With the help of the ASCII – American Standard Code for Information Interchange table taken from Gorn, Bemer & Green (1963), the input plaintext is translated into its equivalent hexadecimal value, which is then given as input. Vivado 2016.2 is used for the simulation.

Plaintext in English: ELECTRONICS

Plaintext in Hex (128 bits): 454c454354524f4e494353

Key in English: PSGCOLLEGEOFTECHNOLOGY

Key in Hex (256 bits): 0505347434f4c4c4547454f465445 43484e4f4c4f4759

The simulation result of the word 'ELECTRONICS' obtained in Vivado 2016.2 using AES 256 algorithm is shown in Figure 4.

The Register Transfer Logic (RTL) schematic of the design generated by Vivado 2016.2 can be seen in Figure 5. The RTL schematic of each round in AES obtained by the synthesis process is given in Figure 6.

FPGA Implementation

Vivado 2016.2 ISE Design Suite is used to carry out the proposed algorithm and the results are targeted on Zybo–Zynq Z-7010 AP SoC development board. The experimental setup for the same is shown in Figure 7.

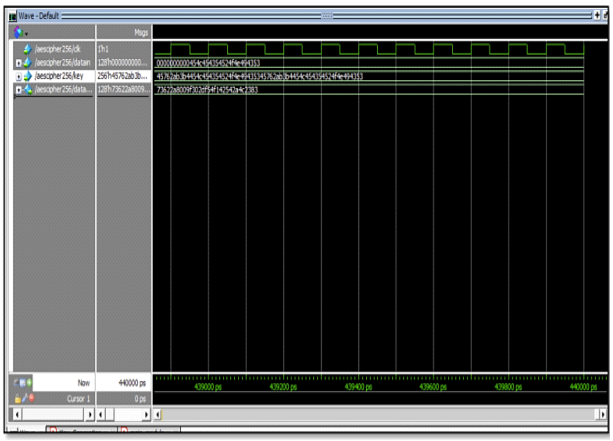

Figure 4: Implementation of proposed AES – 256 for the word 'ELECTRONICS'

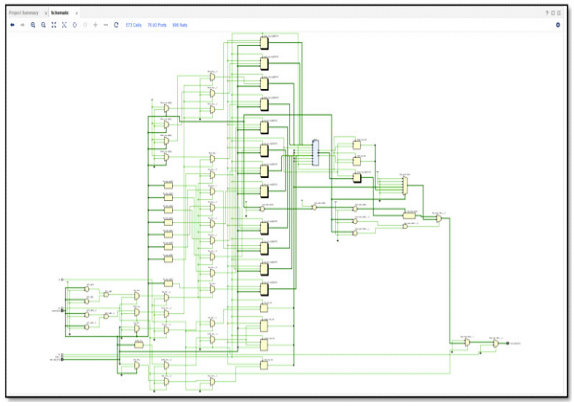

Figure 5: RTL Schematic generated

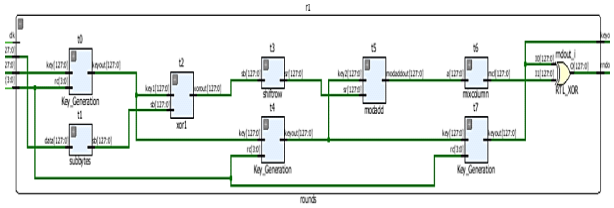

Figure 6: RTL Schematic of each round in AES

Result And Discussion

Power and Utilization Analysis

The total on chip power, junction temperature and resource utilization for the AES algorithm on Zybo-Zynq Z-7010 AP SoC board is tabulated in Table 3.

Because of the additional modules in each round, such as exclusive or modulo addition, and the extra operations in the sub-bytes process due to the additional key generation procedures, the proposed AES uses slightly more power and resources than the conventional AES. However, the increase in diffusion rate of the proposed AES algorithm, which can be proved by avalanche effect, justifies the extra power.

Avalanche Effect

The avalanche effect is used to test the changes made to AES algorithm using 10 different plaintext samples and to estimate the performance of the proposed algorithm in

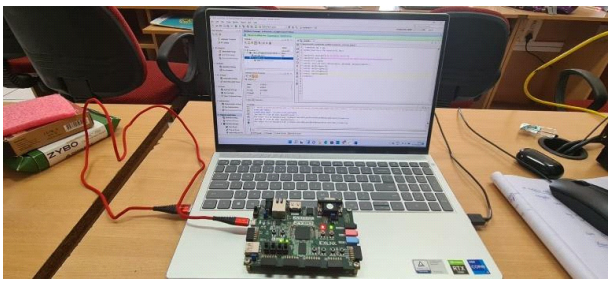

Figure 7: Experimental setup

Table 3: Comparison between conventional and proposed AES algorithm

Parameters	*Conventional AES*	*Proposed AES*
Total on chip Power	1.328 W	1.513 W
Dynamic Power	1.087 W	1.396 W
Junction Temperature	40.5 C	42.5 C
Slice LUTs	7012	7949
F7 Muxes	2008	2480
F8 Muxes	597	689

Table 4: Outputs of Standard AES Encryption for 5 samples

PLAINTEXT	ENCRYPTED PLAINTEXT	PLAINTEXT WITH ONE BIT FLIPPED	ENCRYPTED PLAINTEXT WITH ONE BIT FLIPPED
6AAB3E2CE1EB488AEDE3E5C271E7B59D	6FBDBDBE2AAAF889FE7D9812C7765D82	6AAB3E2CE1EB688AEDE3E5C271E7B59D	5084EAEDE99F2869DD8678976FABAB47
0AE231D3CC3865DAB650465BE5A61D62	6D904E0F746CE9950BB4B7257706BC6F	0AE231D3CC3865DAB650465BE5A61D63	60DAAADCDB8F9A5F1EBD1718BCAF4521
2A60EF8D9F0F6909612E7CE1734F33D6	E46E0D7E8740FBB9D6D765F7ECE61CED	2A60EF8D9F0F692612E7CE1734F33D6	EA62609CD436C42B97C15CA99B6509E9
AFDEAE30C1D4A939E89011467C11C955	F7864F3BC4785C3F306B6C7735A8D632	AFDEAE30C1D4A939689011467C11C955	8ABE1E0CAF227A539A0CCE4625D3B827
AEF6A545B7B00CA10225CBE70EE25907	93737532CA9A374202A85A265879B2AD	AEF6A545B7B00CA14225CBE70EE25907	272A23CF82DA8109678SFFE9EDD646F3

terms of diffusion properties. As a result, a single bit of the plaintext's status is flipped and the hexadecimal character represents this bit status change. The avalanche effect is a term used in cryptography to describe a certain behavior of mathematical functions used for encryption. One of the desirable properties of any encryption scheme is the avalanche effect (i.e.) a small alteration in the plaintext should make the cipher-text to alter significantly (De Los Reyes, Sison and Medina, 2018). To put it another way, it evaluates the effects of a small change in plain text or the key on the cipher text and can be calculated using Eq. 6.1.

$$\text{AVALANCHE EFFECT} = \frac{\text{(Number of Changed bit in ciphertext)}}{\text{(Number of bits in ciphertext)}} \tag{6.1}$$

The plaintexts listed in Table 3 are encrypted using the secret key: PSGCOLLEGEOFTECHNOLOGY. The output of standard AES encrypted cipher text for the 5 sample inputs of the initial plaintexts and one bit altered plaintexts is given in Table 4. The encrypted cipher text results for the proposed AES algorithm are shown in Table 5.

Table 5: Outputs of Modified AES Encryption for 5 samples

PLAINTEXT	M.AES ENCRYPTED PLAINTEXT	PLAINTEXT WITH ONE BIT FLIPPED	M.AES ENCRYPTED PLAINTEXT WITH ONE BIT FLIPPED
6AAB3E2CE1EB488AEDE3E5C271E7B59D	18AA0019AF8B669EECE63AE57155CFE4	6AAB3E2CE1EB688AEDE3E5C271E7B59D	D65DCD48AEEC1E240A653CC5747B67FA
0AE231D3CC3865DAB650465BE5A61D62	BA1E83967AF7699223F6DC125BF4C44D	0AE231D3CC3865DAB650465BE5A61D63	19D905B7B6FA9E2147AFA662767CFEBB
2A60EF8D9F0F6909612E7CE1734F33D6	E68585DC6BBA52013EE676BE0AF2F10E	2A60EF8D9F0F692612E7CE1734F33D6	946C0199AE2EB5241747DF154A05D50C
AFDEAE30C1D4A939E89011467C11C955	BF20B2AAEAF5162BCB6A3736EEC56F75	AFDEAE30C1D4A939689011467C11C955	D3D9CCE19AAE704650C04DB4FB922880
AEF6A545B7B00CA10225CBE70EE25907	65BB3DB5163CA56B388FA44518156F45	AEF6A545B7B00CA14225CBE70EE25907	616C805C353F49F5FDBA163DD92BD668

EXAMPLE	ROUND 1		ROUND 2		ROUND 3		ROUND 4		ROUND 5		ROUND 6		ROUND 7	
	AES	M.AES	AES	M.AES	AES	M.AES	AES	M.AES	AES	M.AES	AES	M.AES	AES	M.AES
1	10	20	63	70	57	73	66	63	66	70	64	65	70	72
2	10	20	65	74	62	66	69	74	67	62	62	65	65	68
3	15	16	75	77	55	67	71	69	62	65	75	77	72	75
4	13	20	59	73	69	72	61	68	63	67	60	65	63	65
5	16	19	62	69	69	84	55	72	76	67	72	75	72	72

Table 6: Avalanche effect

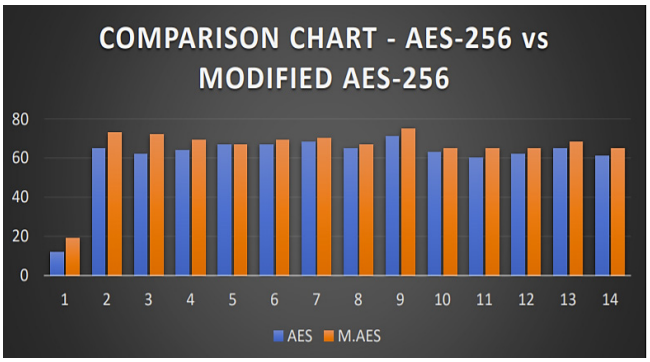

Figure 8: Comparison chart of Avalanche effect of AES – 256 Vs Modified AES – 256

The avalanche effect of both the standard AES and modified AES is tabulated in Table 6 for the first 7 rounds to illustrate the result of the modified AES cipher round. The proposed AES cipher round and key schedule provides a substantially larger avalanche impact than the regular AES, based on these findings. Figure 8 depicts a graphical representation of the same.

Conclusion

As the world is approaching towards more automated information resources, encryption will become more important as a security measure. Access control and data security will be strengthened in electronic networks for banking, inventory control, retail, distributed processing, benefit and then service delivery, information storage and for retrieval and government applications. Using the cryptography technology, information security can be simply done.

The modifications done to the existing system are expected to enhance the security level and make the process in much simpler and faster way. According to the results, the diffusion characteristics of the proposed AES have improved in both the early and full rounds of the encryption part. This enhancement is accounted to the use of added elementary operations like exclusive OR, modulo arithmetic and also the modifications introduced to the sub-bytes operation which introduces more key variations in cipher round.

Based on the analysis, the proposed AES methodology provides a larger avalanche effect than the conventional AES. Thus, an algorithm is proposed that is more secure and efficient in securing confidential information in Internet of Things (IoT) environment.

References

Borkar, A. M., Kshirsagar, R. V., & Vyawahare, M. V. (2011). FPGA implementation of AES algorithm. 3rd International Conference on Electronics Computer Technology, 3, 401-405. IEEE.http://toc.proceedings.com/12007webtoc. pdf

De Los Reyes, E. M., Sison, A. M., & Medina, R. (2019). Modified AES cipher round and key schedule. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 7(1), 29-36. doi:10.1109/ICIIBMS.2018.8549995.

Deshpande, H. S., Karande, K. J., & Mulani, A. O. (2014). Efficient implementation of AES algorithm on FPGA. International Conference on Communication and Signal Processing, 1895-1899. IEEE. doi: 10.1109/ ICCSP.2014.6950174

Elmogy, A., Bouteraa, Y., Alshabanat, R., & Alghaslan, W. (2019). A New Cryptography Algorithm Based on ASCII Code. 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), 626-631. IEEE. doi: 10.1109/STA.2019.8717194

Gorn, S., Bemer, R. W., & Green, J. (1963). American standard code for information interchange. Communications of the ACM, 6(8), 422-426.

Gupta, R.K. (2020). A Review paper on concepts of cryptography and cryptographic hash function. Eur J Mol Clin Med, 7(7), 3397-408.

Kumar, K., Ramkumar, K. R., & Kaur, A. (2020). A design implementation and comparative analysis of advanced encryption standard (AES) algorithm on FPGA. 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 182-185. IEEE. doi: 10.1109/ ICRITO48877.2020.9198033.

Parikh, P., & Narkhede, S. (2016). High performance implementation of mixing of column and inv mixing of column for AES on FPGA.International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC),174-179. IEEE. doi: 10.1109/ ICCPEIC.2016.7557244.

Setetemela, K. O., Keta, K., Nkhabu M. and Winberg, S. (2019). Python-based FPGA implementation of AES using Migen for Internet of Things Security, IEEE 10th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT). 9, 194-198. doi: 10.1109/ ICMIMT. 2019.8712074.

Soliman, S. M., Magdy, B. and Abd El Ghany, M. A. (2016). Efficient implementation of the AES algorithm for security applications. 29th IEEE International System-onChip Conference (SOCC), 206-210. doi: 10.1109/ SOCC.2016.7905466.

Su, N., Zhang, Y., & Li, M. (2019). Research on data encryption standard based on AES algorithm in internet of things environment. IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2071-2075. IEEE. doi: 10.1109/ ITNEC.2019.8729488

Talirongan, H., Sison, A. M., & Medina, R. P. (2018). Modified advanced encryption standard using butterfly effect. In 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 1-6. IEEE. https://doi.org/10.1109/HNICEM.2018.8666368

Wang, W., Chen, J., & Xu, F. (2012). An implementation of AES algorithm Based on FPGA. 9th International Conference on Fuzzy Systems and Knowledge Discovery,1615-1617. IEEE. doi: 10.1109/FSKD.2012.6233811.

Wood, C. C. (1981). Future Applications of Cryptography. IEEE Symposium on Security and Privacy, 70-70, doi: 10.1109/ SP.1981.10013.

Yuan, Y., Yang, Y., Wu, L., & Zhang, X. (2018). A high performance encryption system based on AES algorithm with novel hardware implementation. IEEE International Conference on Electron Devices and Solid State Circuits (EDSSC), 1-2. IEEE. doi: 10.1109/EDSSC.2018.8487056.