Int. j. inf. tecnol. (September 2020) 12(3):825–830
https://doi.org/10.1007/s41870-020-00465-1

ORIGINAL RESEARCH

A novel method of improvement in advanced encryption standard algorithm with dynamic shift rows, sub byte and mixcolumn operations for the secure communication

T. Manoj Kumar[1] · **P. Karthigaikumar**[1]

Received: 2 November 2019 / Accepted: 23 April 2020 / Published online: 6 May 2020

Abstract In this digital information era, data transmitted over internet is vulnerable to several types of attacks. To preserve our information over internet, it is necessary to use some cryptographic methods. advanced encryption standard is a strong symmetric, non-fiestel cryptographic algorithm which preserves the information transmitted in internet from piracy or attacks. In this research article, a new and an efficient key dependent AES algorithm is presented for securing the data over internet or digitally stored in any remote location. This proposed work gives better avalanche effect and strict avalanche effect on comparing with the original existing AES algorithm. Results of the proposed algorithm are obtained after testing 1000 pairs of samples of different secret key. And it shows that 58% of the times, new proposed algorithm have better avalanche effect than the existing algorithm.

Keywords Advanced encryption standard · Encryption · Decryption · S-box · Mix column · Affine transformation · Shift rows · Key scheduling · MSC-94 02 · MSC-94 04 · MSC-94 06

1 Introduction

In 2001, National Institute of Standards and Technology (NIST) published AES as Federal Information Processing 197 (FIPS 197) [1]. AES was originally designed by Belgian researchers, Joan Daemen and Vincent Rijment. AES is a symmetric cryptographic algorithm which uses same secret key for both encryption and decryption operation. AES process data of block size 128 bits with different key sizes, 128,192, 256 bits. To get cipher text through AES algorithm plain text has to go through several rounds of AES process. Key size determines the number of rounds plain text has to be processed. AES algorithm with Key size 128 uses 10 rounds, 192 uses 12 rounds and 256 uses 14 rounds [2–6].

Due to the tremendous growth of technology, it is still possible to decode the 8 rounds in a 10 round AES. Remaining 2 rounds can also be brute forced with sufficient information on either plain text or secret key. Therefore to make AES as strong cryptographic algorithm it is necessary to modify the architecture of the existing algorithm. In this research work, the entire process in the encryption and decryption are made as secret key dependent. The following sections in this article is organized as follows Sect. 2 details the related research works of this research article, Sect. 3 explains the overview of the AES algorithm. In Sect. 4, proposed algorithm is explained in details, whereas in Sect. 5 has experimental results and its explanation and conclusion of this research is given in Sect. 6.

2 Related researches

Wadi et al. [7] given an approach to increase the speed of the execution of the AES algorithm by reducing the number of rounds of operation on plain text to obtain a single block The above Fig. 2 gives the simulation result of the proposed dynamic AES algorithm. Output is shown for two different keys with same plain text. Keys used for this simulation differ by a single bit and output can be measured for the avalanche effect analysis of cipher text.

✉ T. Manoj Kumar
srimanojkumar@gmail.com

[1] Karpagam College of Engineering, Coimbatore, India

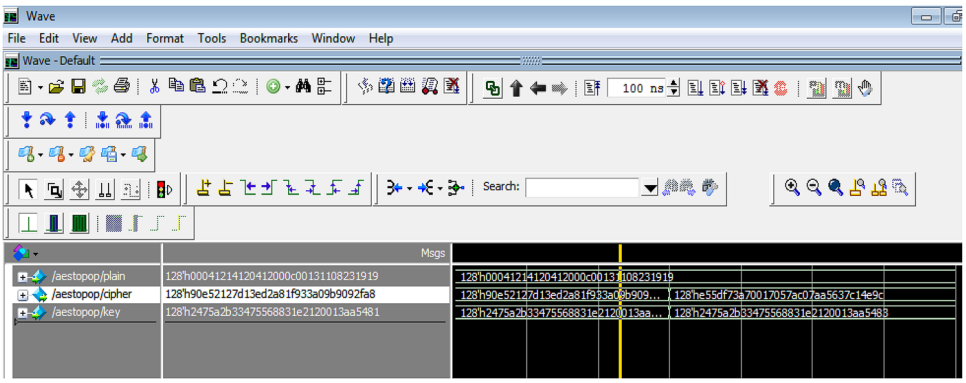

Fig. 1 Simulation result of proposed dynamic AES algorithm

Zhang et al. [8] proposed composite field arithmetic in s-box operation which increases the speed of the operation but also reduces the complexity of the s-box operation.

Rahimunnisa et al. [9] uses parallel processing and folded architecture to obtain high throughput.

Granado Criado et al. [10] designed an AES-128 algorithm with parallelism, pipelining, partial and dynamic reconfiguration. With new architecture a high th5roughput is achieved.

Jyrwa et al. [11] have worked with an iterative architecture and modifications including merging.

of SubBytes and ShiftRows, LUTs. This architecture yields throughput of 1.458 Gbps.

Sumio Morioka et al. [12] gives a modified S-Box circuit with low power PPRM architecture. This architecture reduces the power consumption by reducing the dynamic hazards and achieves lower consumption of 29 μW at 10 MHz.

Koraida et al. [13] described the performance of the modified AES algorithm by replacing the generalized matrix multiplication with the permutation table used in the DES process. With this modification, encryption and decryption becomes much faster than the existing algorithm.

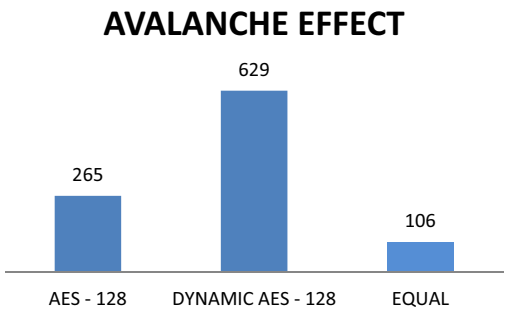

Fig. 2 Comparison of avalanche effect

Oyshee et al. [14] described an alternate approach for different s-box design with different Affine transformation process.

3 AES algorithm overview

In AES encryption and decryption operation the length of the input block, state and output block is 128 bits. Length of the secret key is 128,192 and 256 bits. Number of rounds of AES operation to be performed on a single block of data is determined by the key size. AES operations are performed as matrix calculation. Input plain text and subkey are transformed into matrix before being inserted into AES stages [15–19] (Table 1).

For both encryption and decryption rounds of operation AES uses four byte oriented transformations.

1. Sub byte transformation/inverse sub byte transformation
2. Shift rows/ inverse shift rows
3. Mix column/inverse mix column
4. Add round key

3.1 Sub byte transformation/inverse sub byte transformation

It is a non-linear byte to byte transformation technique, obtained as result of multiplicative inverse operation followed by affine transformation.

Table 1 AES Specification

Algorithm	Key length	Block size	No of rounds
AES-128	128	128	10
AES-192	192	128	12
AES-256	256	128	14

3.2 Shift rows

In this transformation, bytes in the state are shifted cyclically left shifted over different numbers for different rows. For first row, bytes are not rotated, where as second row shifted once, third row two times and fourth row is shifted three times byte by byte.

3.3 Mix column

This transformation technique is operated column by column in a state matrix. Output of the mix column technique is obtained after performing the multiplication of state matrix with the standard mix column matrix.

For Mix column operation

$$\begin{bmatrix} S'1 \\ S'2 \\ S'3 \\ S'4 \end{bmatrix} = \begin{bmatrix} 02030101 \\ 01020301 \\ 01010203 \\ 03010102 \end{bmatrix} \begin{bmatrix} S1 \\ S2 \\ S3 \\ S4 \end{bmatrix} \quad (1)$$

For Inv Mix column operation

$$\begin{bmatrix} S'1 \\ S'2 \\ S'3 \\ S'4 \end{bmatrix} = \begin{bmatrix} 0e0b0d09 \\ 090e0b0d \\ 0d090e0b \\ 0b0d090e \end{bmatrix} \begin{bmatrix} S1 \\ S2 \\ S3 \\ S4 \end{bmatrix} \quad (2)$$

3.4 Add round key

This is the only operation in which the secret gets operated with the pain text. A simple EXOR operation is carried out between the output of the mix column and subkey of that particular round.

3.5 Key scheduling

Key scheduling or Key expansion is a process of obtaining required number of subkeys one for each round of operation from the main secret key. Each subkey is dependent on the previous subkey. It's a very important process, since there should not be any resemblance between any subkey. It involves word rotation, word shifting, addition of a constant value, EXOR operation [20–26].

4 Proposed AES technique

From the previous section, it is clear that the secret is used only in the add round key section. All the other sections are independent of secret key. Therefore those operations are static and can be easily revertible. To avoid this, a much more efficient AES algorithmic technique in which are the operations are performed based on the key. In shortly, all sections produce the result based on the key. Due to this property, operations on each round of proposed AES is dynamic.

4.1 Dynamic S-box transformation

S-box in the proposed work will not be same for all the rounds of AES operation. Therefore it is called as dynamic s-box depending on the key and subkey of that particular round of operation in AES. Because of this dynamic s-box, it is not easy for a third party attacker to crack the s-box without the information about the secret key.

In this work, S-box is made key dependent by the two following steps:

(i) EXOR operation is performed between the byte values of the secret key or subkey of that particular round to find a resultant value.
(ii) Based on the resultant value, the standard s-box value is rotated cyclically left. That rotated value is used as the replacement for the input value of the s-box stage.

For example: byte wise EXOR operation in the key b955b5ce8d20e346bcc2f146af68a5c1 gives F5. Therefore standard s-box values are rotated cyclically left F5 times to provide the final result. Input of 14 to the s-box stage gives 5F as the result in the proposed work instead of FA.

During decryption operation, s-box values are rotated cyclically right by the resultant value obtained as the result of byte wise EXOR operation in the subkey to give the desired output.

Pseudo-code for dynamic s-box transformation

```
Pseudo-code for Dynamic S-Box Transformation
SubBytes(E)
{
d = E-1
smatrix (d,B)
for (j = 0 to 7)
{
cj = Bj ^ B(j+4)mod8^B(j+5)mod8^B(j+6)mod8^B(j+7)mod8
Fj = cj^smatrix(0X63)
}
smatrix-1(F,f)
A = EXOR(bytes)
G = f<<A
}
```

Table 2 Comparison of avalanche effect between standard AES and proposed dynamic AES

Plain text	Secret key	AES-128(1)	No of bits change	Avalanche effect	Dynamic AES-128	No of bits change	Avalanche effect
00041214120412000c00131108231919	2475a2b33475568831e2120013aa5481	23afddb8de431de6f27d202e237cecae	59	0.46	90e52127d13ed2a81f933a09b9092fa8	67	0.52
	2475a2b33475568831e2120013aa5483	93be7767af0a2a6a40f4c258ab48268d			e55df73a70017057ac07aa5637c14e9c		
	2475a2b33475568831e2120013aa5486	8d47144ad799afcc3b6139cf67154b2d1	65	0.507	4a26015dffc5168a86ab3710cd3bdc9c	73	0.57
	2475a2b33475568831e2120013aa5487	bc028bd3e0e3b195550d6df8e6f18241			3604d6275969ad6353daefbfd08300af		
	2475a2b33475568831e2120013aa548f	92e27ea9c842ff2c5f53315e6525442e	53	0.41	502efec916cbfadfd7311621f25fd621c	64	0.5
	2475a2b33475568831e2120013aa548e	57f18d13a980bf50d734f1c6a449407a			a19caa62e36ef4b8edcb2c29a9fe2a43		
	2475a2b33475568831e2120013aa548c	56a954c9141807e05e81023b413d8ec2	74	0.578	b16cc3a6cb9b0ef75cfe844c0624d3df	74	0.578
	2475a2b33475568831e2120013aa548d	1b5e912c00b29aee8b5ef2c7387aa1a8			eef0ad9c4c3d7116aab43f7149b75c0f		

4.2 Dynamic shift rows

In the standard Shift row technique, state matrix's second, third and fourth rows are shifted cyclically left one, two, three times respectively. This technique is also static or same for the entire different types of subkey. In this proposed work, rows of the state matrix are shifted according subkey information. This technique is implemented in three steps.

(i) Parity of the subkey of that particular round is identified by having the EXOR operation between all the bits of the subkey
(ii) If the parity bit is '1′, second and fourth rows are shifted cyclically left by one and three times respectively.
(iii) If the parity bit is '0′, second and third rows are shifted cyclically left by one and two times respectively.

Because of this property in this new technique, Shift row operation becomes dynamic based on the subkey parity value.

In decryption, the state matrix rows are rotated cyclically as same as in encryption, except rows are rotated right instead of left rotation.

4.3 Dynamic Mix column transformation

Like the previous stages of AES operation, In Mix column operation the matrix used to multiply the input is same for all rounds irrespective of the subkey. In this proposed work, a slight change is given to the mix column stage. Two matrixes are used to perform this operation. This process is implemented in three steps.

(i) Parity value of the subkey of that particular round is calculated by applying EXOR operation between all the bits of the subkey
(ii) If the parity value is '1′, the following matrix is used for mix column operation

$$\begin{bmatrix} S'1 \\ S'2 \\ S'3 \\ S'4 \end{bmatrix} = \begin{bmatrix} 02030101 \\ 01020301 \\ 01010203 \\ 03010102 \end{bmatrix} \begin{bmatrix} S1 \\ S2 \\ S3 \\ S4 \end{bmatrix} \quad (3)$$

(iii) If the parity value is '0′, the following matrix is used for mix column operation

Table 3 Comparison of avalanche effect

Number of samples	AES-128	Dynamic AES-128	Equal
1000 pairs	265	629	106

Table 4 Synthesis report

Parameters	SubByte		Shift row		MixColumn		Overall	
	AES-128	This work	AES-128	This work	AES-128	This work	AES-128	This work
Power	29	22	106	112	72	78	560	435
Area	38	32	256	286	152	182	9599	18,326

$$\begin{bmatrix} S'1 \\ S'2 \\ S'3 \\ S'4 \end{bmatrix} = \begin{bmatrix} 0e0b0d09 \\ 090e0b0d \\ 0d090e0b \\ 0b0d090e \end{bmatrix} \begin{bmatrix} S1 \\ S2 \\ S3 \\ S4 \end{bmatrix} \quad (4)$$

During decryption operation, matrix used in mix column gets interchanged for the generated parity value. For parity '1' Eq. 2 is used and for parity '0′ Eq. 1 is used for performing inverse mix column operation.

5 Experimental results

The above modified new dynamic AES algorithm is designed in verilog and verified its functionality using Modelsim and implemented in xilinx FPGA device xc6vlx75t-3ff784. Table 2 gives the simulation result of both standard AES algorithm and the proposed dynamic AES algorithm. Comparison between the cipher texts obtained on the simulation both the algorithms for a single bit change in the secret key is also provided in Table 3. Plain text is kept constant for the AES encryption operation. A single bit change is induced in the secret to measure the avalanche effect and check whether the new algorithm satisfies the strict avalanche criterion (Figs. 1, 2).

5.1 Simulation result

5.1.1 Avalanche effect

For any cryptographic algorithm, the most desired property is Avalanche effect (4). It can be measured by changing a single bit in the input value and measuring the bits changes in the output for a single bit input change. Comparison between two outputs is made to determine the avalanche effect of the algorithm.

$$\text{Avalanche effect} = n_b/t_b \quad (5)$$

where, n_b = total number of bits changed in output. t_b = total number of bits in output.

From the above Eq. 5 it is clear that the avalanche effect is directly proportional to the number bits changed in output due to a single bit change in input. Therefore high avalanche effect determines the large number of bits changed in output.

A cryptographic algorithm with high avalanche effect is harder for an attacker to crack the encryption process (5). It is wise to choose the algorithm which has high avalanche effect, to ensure that even a minute change has big impact on the output of the algorithm. For our proposed algorithm, 2000 samples of secret key are used for testing the avalanche effect. In this 1000 samples are differ in one bit on comparing with the other 1000 samples of key. Plain text remains same throughout the testing process. Because of the modification induced in this proposed algorithm, about 629 times the number of bits changed in the output is high in the proposed algorithm. About 265 times the number of bits changed in the output is high in the standard AES algorithm whereas, number of bit changes remains same in the both the algorithm is 106 times. Therefore our proposed algorithm is much more efficient since, it exhibits high avalanche effect when compared with the standard algorithm. Also from Table 2, it is clear that almost 50% of the bits get change in the output while employing proposed dynamic AES algorithm satisfying strict avalanche criterion. Table 3 and Fig. 2 gives the comparison result of avalanche effect between AES – 128 and our proposed work with 1000 pairs of samples are taken for the analysis.

5.1.2 Synthesis result

Above Table 4 gives the synthesis result of proposed algorithm. From the result it is clear that with sacrifice in the area, it is possible to achieve a logic circuit for which consumes low power as well as high avalanche effect.

6 Conclusion

This article concentrates on improving the security of information in the UAV by implementing dynamic AES-128 algorithms. Different versions of AES have been designed are more efficient than the standard AES algorithm in terms of security other parameters. Some of the algorithms provide better performance in terms of area, power consumption and time complexity. With this newly proposed dynamic AES algorithm, the security of the

cryptographic algorithm is enhanced by providing greater avalanche effect than the standard AES. By providing the key dependency in all the operations of AES, the avalanche effect has been increased in the output. About 58% increment has taken place in the avalanche effect in the proposed AES algorithm on comparing with the standard AES algorithm. This research article aims to improve only the avalanche effect on the output rather than concentrating area, time and power consumption by the proposed architecture.

References

1. National Institute of Standards and Technology. Advanced Encryption Standard (AES). Federal Information Processing Standards Publications—FIPS 197. http: //csrc.nist.gov/publications/fips/ fips197/fips-197.pdf. November 2001
2. Daemen J, Rijmen V (2002) The design of Rijndael. Springer, Berlin
3. Stallings W (2014) Cryptography and network security, 5th edn. Pearson Education Inc, London
4. Juremi J et al (2012) Enhancing advanced encryption standard s-box generation based on round key. International Journal of Cyber-Security and Digital Forensics 1(3):183–188
5. Krishnamurthy GN et al (2011) Study of effect of removal of shiftrows and mixcolumns stages of AES and AES-KDS on their encryption quality and hence security in international scholarly and scientific research & innovation 5(2):143–156
6. Mathew SK, Sheikh F, Kounavis M, Gueron S, Agarwal A, Hsu SK et al (2011) 53 Gbps GF(2^4) native composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high performance microprocessors. IEEE J Solid State Circuits 46(4):767–776
7. Wadi SM, Zainal N (2013) Rapid encryption method based on AES algorithm for grey scale HD image encryption in Elsevier. In: Proceedings of the 4th international conference on electrical engineering and informatics (ICEEI 2013), vol 11, pp 51–56
8. Zhang X, Parhi KK (2006) On the optimum constructions of composite field for the AES algorithm. IEEE Trans Circuits Syst 53(10):1153–1157
9. Rahimunnisa K, Karthigaikumar P, Rasheed S, Jayakumar J (2012) FPGA implementation of AES algorithm for high throughput using folded parallel architecture. Secur Commun Netw 7(11):2225–2236
10. Granado Criado JM, Vega Rodriguez MA, Sanchez Perez JM, Gomez Pulido JA (2010) A new methodology to implement the AES algorithm using partial and dynamic reconfiguration. Integr VLSI J 43:72–80
11. Jyrwa B, Paily R (2009) An area-throughput efficient FPGA implementation of block cipher aes algorithm. In: International conference on advances in computing, control and telecommunication technologies, pp 328–332
12. Morioka S, Satoh A (2003) An optimized s-box circuit architecture for low power AES design. Springer, Berlin
13. Koradia VC et al (2013) Modification in advanced encryption standard. J Inf Knowl Res Comput Eng 2(2):356–358
14. Oyshee B et al (2012) An optimized s-box for advanced encryption standard (AES) design. In: The proceedings of 2012 international conference on advances in computing and communications, pp 154–157.
15. Manoj Kumar T, Karthigaikumar P (2018) FPGA implementation of an optimized key expansion module of AES algorithm for secure transmission of personal ECG signals. Des Autom Embed Syst 22(1–2):13–24
16. Manojkumar T, Karthigaikumar P, Ramachandran V (2019) An optimized s-box circuit for high speed AES design with enhanced PPRM architecture to secure mammographic images. J Med Syst 43:31
17. Karthigaikumar P, Christy NA, Mangai NMS (2015) PSP CO2: an efficient hardware architecture for AES algorithm for high throughput. Wirel Personal Commun 85(1):305–323
18. Rahimunnisa K, Karthigaikumar P, Christy N, Kumar S, Jayakumar J (2013) PSP: parallel sub-pipelined architecture for high throughput AES on FPGA and ASIC. Eur J Comput Sci 3(4):173–186
19. Mangard S, Aigner M, Moninikus S (2003) A highly regular and scalable AES hardware architecture. IEEE Trans Comput 52(4):483–491
20. Abuelyman ES, Alsehibani AAS (2008) An optimized implementation of the s-box using residue of prime numbers. Int J Comput Sci Netw Secur 8(4):304–309
21. Itoh T, Tsujii S (1988) A fast algorithm for computing multiplicative inverses in GF(2m) using normal bases. Inf Comput 78(3):171–177
22. Good T, Benaissa M (2006) Very small FPGA application-specific instruction processor for AES. IEEE Trans Circuits Syst I Regul Pap 53(7):1477–1486
23. Good T, Benaissa M (2010) 692-nW advanced encryption standard (AES) on a 0.13-m CMOS. IEEE Trans Very Large Scale Integr VLSI Syst 18(12):1753–1757
24. Huang J, Lai X (2012) Transposition of AES key schedule. Int Assoc Cryptol Res 260:1–13
25. Paul A, Victoire TAA, Jeyakumar AE (2003) Particle swarm approach for retiming in VLSI. In: IEEE 46th midwest symposium on 3 circuits and systems, 2003, pp 1532–1535.
26. Helion. www.heliontech.com.