Implementation of 128-bit AES algorithm in MATLAB

D.Lohit Kumar[1] Dr. A.R.Reddy[2] Dr.S.A.K.Jilani[3]
[1]*PG Scholar* [2]*Professor* [3]*Professor, Dept. of ECE*
MITS, Madanapalle, INDIA

Abstract:

Advanced Encryption Standard (AES) is used for securing data. There are several types of algorithms available in cryptography, but AES is one among the standardized high security algorithm. It is implemented in various hardware devices and various software languages. This paper explores the implementation of AES in MATLAB. A MATLAB code is developed for plaintext encryption and cipher text decryption. Experiments are conducted to measure the execution times. The measured results indicate a maximum encryption time of 87 ms and decryption time of 88 ms. These results are superior to the similar software implementations of AES. The results are attractive to deploy AES into image and video processing on MATLAB.

Keywords*:* *Advanced Encryption Standard, symmetric key Encryption, cipher text and MATLAB, internet, multimediafiles.*

I. INTRODUCTION

With the advent of internet and multimedia files many old encryption algorithms are finding new platform for their application. Rijndael algorithm is adopted as Advanced Encryption Standard by NIST [1]. The AES algorithm uses a round function that is composed of four different byte-oriented Layers. They are byte substitution, shifting rows, mix columns, add around key. In AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by Nr, where Nr=10 for 128 bits, Nr= 12 for 192 bits and N_r= 14 for 256 bits.

AES is already implemented in various processors including microcontrollers [11]. Also, AES is implemented in many platforms like ASIC [9], FPGA [5], Embedded systems like Arduino, Raspberry Pi.

This paper explores the implementation of AES in MATLAB. It uses a high level language and useful to develop many applications such as video processing, control systems, signal processing and communications. Therefore, development of AES in MATLAB will be useful for many applications which required data protection. For example, encryption of image will be very useful to ensure the security.

This paper is organised as follows: Section II covers introduction of AES algorithm. Section III discusses the implementation of AES algorithm using MATLAB. Section IV explains about simulation of AES in MATLAB. Section V explains about experimental results and analysis.

II.AES algorithm

AES algorithm is a Block cipher and works on fixed-length group of bits, which are called Blocks. It takes an input block of 128 bits, and produces a corresponding output block of the same size. AES requires a second input, which is the secret key. AES uses three different key sizes: 128, 192 and 256 bits.

For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is composed of four different byte-oriented Layers:

1) Byte substitution using a substitution table (S-box)
2) Shifting rows of the State array by different offsets
3) Mixing the data within each column of the State array
4) Adding a Round Key to the State.

AES algorithm uses this round function for ten rounds, but the first and last round of AES algorithm is differing from other rounds. That is depicted in the figure 1 below.

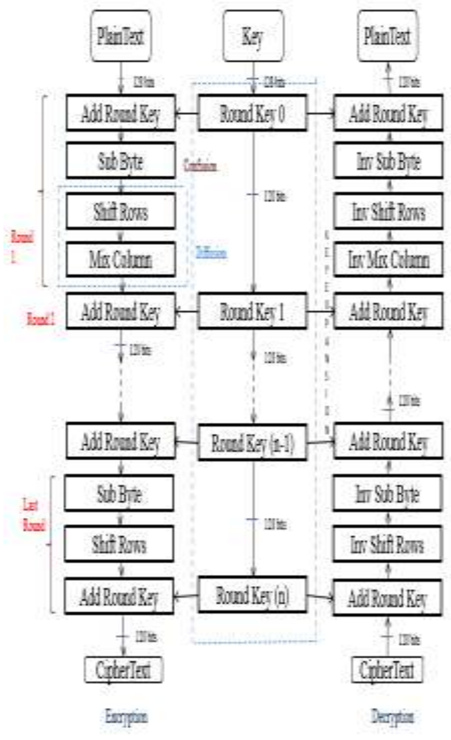


Fig.1: AES Algorithm Architecture

In this AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by Nr, where Nr=10 for 128bits, Nr= 12 for 192bits and N_r= 14 for 256bits.

III. MATLAB platform used for AES

MATLAB is a high level language and a comfortable environment for numerical calculations, visualization and programming. Using this platform we can create models, develop algorithms, and applications. Using built in math functions of this platform enables us to achieve the result faster than spreadsheets or conventional programming languages, like C/C++ or Java.

With the help of MATLAB compiler we can generate executable code for any computer system for different environments.

With the advancement of VLSI technology single board computers are dominating in the markets. The codes deployed were made run in the systems.

This platform has wide range of applications like signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology.

Take a new script and write the program into new script and save the file in destination folder which is required for AES encryption and decryption.

In the first step Initialisation of AES components are done like S-box and Inv S-box are created then poly matrices are created and in the second step round function command is used the below commands. Then the shift rows, mix columns, cipher, encryption and decryption pseudo codes are written.

```
function (S-BOX, Inv(S-BOX), W, POLY-MAT,
INV(POLY-MAT)] = AES(INIT)

function (S-BOX, INV(S-BOX)) = S-BOXgen;

function STATE(Out) = Add(RoundKey),
(STATE(in))
STATE(out) = Bitxor(State(in),Roundkey);


function STATE(out) = ShiftRows (STATE(in))
STATE(out) = Cycle(STATE(in));

function Cipher-Text = Cipher (PLAIN-TEXT, W,
S-BOX, Poly-Mat)
Cipher-text = Cipher[PLAIN-TEXT, W, S-BOX,
POLY-MAT];

Re-Plaintext = Inv-Cipher[Cipher-Text, W, Inv[S-
BOX, Inv[Poly-Mat,]]];
PLAIN-TEXTtoHEX = DectoHex(Re-Plain-Text);
```

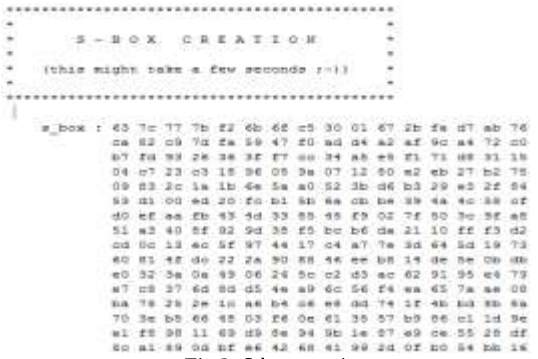
These functions are performed in the MATLAB for AES encryption and decryption process, this involves the S-box and Inv S-box creation, and four byte oriented operations.

Plaintext is given as input by using symmetric key and using above byte oriented operations in the MATLAB, the AES algorithm is implemented.

So that encryption and decryption is done separately.

IV. Simulation of AES in MATLAB platform

In the initialization step the s-box, inv s-box, polynomial matrices are created.



```
*******************************************
*                                         *
*      S - B O X   C R E A T I O N        *
*                                         *
*   (this might take a few seconds ;-))   *
*                                         *
*******************************************

s_box : 63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ab 76
        ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 72 c0
        b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 31 15
        04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 b2 75
        09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 2f 84
        53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c 58 cf
        d0 ef aa fb 43 4d 33 85 45 f9 02 7f 50 3c 9f a8
        51 a3 40 8f 92 9d 38 f5 bc b6 da 21 10 ff f3 d2
        cd 0c 13 ec 5f 97 44 17 c4 a7 7e 3d 64 5d 19 73
        60 81 4f dc 22 2a 90 88 46 ee b8 14 de 5e 0b db
        e0 32 3a 0a 49 06 24 5c c2 d3 ac 62 91 95 e4 79
        e7 c8 37 6d 8d d5 4e a9 6c 56 f4 ea 65 7a ae 08
        ba 78 25 2e 1c a6 b4 c6 e8 dd 74 1f 4b bd 8b 8a
        70 3e b5 66 48 03 f6 0e 61 35 57 b9 86 c1 1d 9e
        e1 f8 98 11 69 d9 8e 94 9b 1e 87 e9 ce 55 28 df
        8c a1 89 0d bf e6 42 68 41 99 2d 0f b0 54 bb 16
```

Fig.2: S-box creation

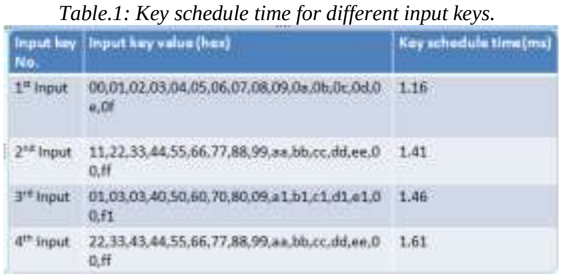
The plaintext of size 128bits are given as input using a symmetric key enables to perform four byte oriented operations to get the cipher text. The four byte oriented operations involves substitution of bytes, shifting rows, mixing of columns, and add-round key. This four byte oriented operations undergoes 10 rounds for simulating 128bit input plaintext and gives the cipher text out so, encryption is done and then cipher text is given as input and then cipher-text is given as input to the decryption process then it performs the inverse process for 4-byte oriented operations and then it undergoes ten rounds for getting the plaintext as decrypted output. The number of rounds depends on the size of the key used in this process. So that encryption and decryption are evaluated and plotted. The key schedule time for this encryption and decryption is being tabulated in the experimental results section. The time of encryption and decryption for one round is also tabulated in the next section. All this entire process is done in HP desktop 64-bit operating system with 4GB-RAM, i5 processor and 2.20 GHz frequency.

V. Experimental Results and Analysis

A block of pain text data is given as input to the AES encryption algorithm. The resulting cipher text is given as input to AES decryption algorithm. A master key of size 128 bit is used by both encryption and decryption algorithms. Experiments are conducted to find out key set up time, one round encryption time and full encryption and decryption time.

A.Key setup time: This time is measured by using *tic toc* command available in the MATLAB. This command used at the beginning and end of the key schedule code. The difference in time period indicates the key setup time. The measured results are tabulated in Table 2. The measured key set up time is ranging from 1.16 ms to 1.61 ms corresponding to various master key values.

Table.1: Key schedule time for different input keys.



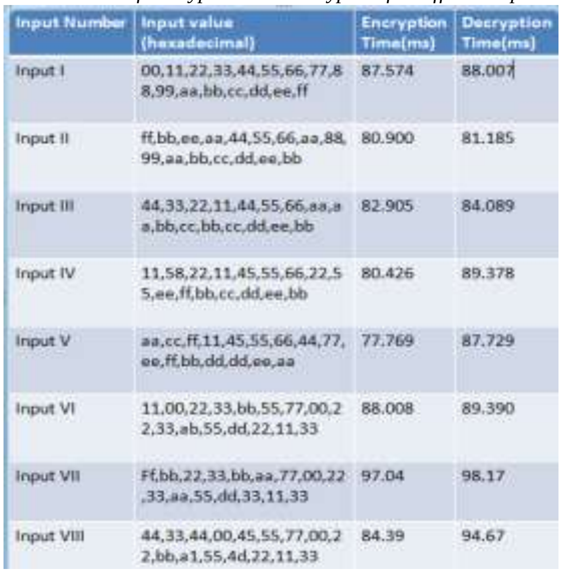
Input key No.	Input key value (hex)	Key schedule time(ms)
1st Input	00,01,02,03,04,05,06,07,08,09,0a,0b,0c,0d,0e,0f	1.16
2nd Input	11,22,33,44,55,66,77,88,99,aa,bb,cc,dd,ee,00,ff	1.41
3rd Input	01,03,03,40,50,60,70,80,09,a1,b1,c1,d1,e1,00,f1	1.46
4th Input	22,33,43,44,55,66,77,88,99,aa,bb,cc,dd,ee,00,ff	1.61

B.One round encryption time*:* This is measured using the method described in key setup time measurement. The *tic toc* command is placed at the beginning and at the end of the first round. The measured encryption time is 12.56 ms for the first round.

C.Full encryption time: This is measured using the method described in the one round encryption time measurement. The *tic toc* command used in the beginning and the end of the main program. The measured encryption time is 87.57ms for first plaintext input shown in the Table 1. Likewise for different eight inputs corresponding encryption times are tabulated.

D.Full decryption time: This is measured using the method described in the one round encryption time measurement. The *tic toc* command used in the beginning and the end of decryption main program. The measured decryption time is 88.007ms for plaintext input shown in the Table 2. Likewise for different eight inputs corresponding decryption times are tabulated.

Table.2: Time of Encryption and Decryption for different inputs.



Input Number	Input value (hexadecimal)	Encryption Time(ms)	Decryption Time(ms)
Input I	00,11,22,33,44,55,66,77,88,99,aa,bb,cc,dd,ee,ff	87.574	88.007
Input II	ff,bb,ee,aa,44,55,66,aa,88,99,aa,bb,cc,dd,ee,bb	80.900	81.185
Input III	44,33,22,11,44,55,66,aa,aa,bb,cc,bb,cc,dd,ee,bb	82.905	84.089
Input IV	11,58,22,11,45,55,66,22,55,ee,ff,bb,cc,dd,ee,bb	80.426	89.378
Input V	aa,cc,ff,11,45,55,66,44,77,ee,ff,bb,dd,dd,ee,aa	77.769	87.729
Input VI	11,00,22,33,bb,55,77,00,22,33,ab,55,dd,22,11,33	88.008	89.390
Input VII	Ff,bb,22,33,bb,aa,77,00,22,33,aa,55,dd,33,11,33	97.04	98.17
Input VIII	44,33,44,00,45,55,77,00,22,bb,a1,55,4d,22,11,33	84.39	94.67

Here, the time of decryption is greater than the time of encryption.

For the above time of encryption and decryption table the corresponding bar chart is drawn and plotted in the figure 3.

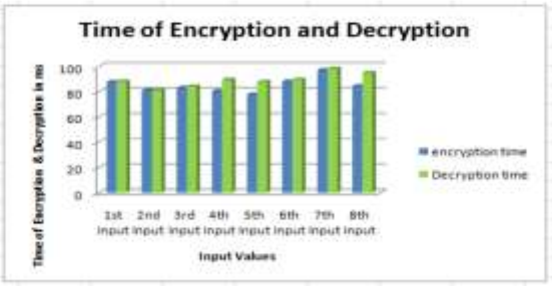


Fig.3: Bar chart for time of encryption and decryption

The above graph shows the time of encryption and decryption, blue colour indicates the encryption time and green colour indicates decryption time.

On the x-axis the input values and on y-axis time of encryption and decryption is taken, first input has its own encryption and decryption time, like wise all for 8-inputs the corresponding outputs are plotted.

E.Results and discussion: The measured results presented above are for MATLAB implementation of AES on HP notebook 64bit OS, i5 processor. These results are compared with the results published by Sambasiva Reddy et al [2]. The MATLAB implementation is able execute within 88 ms, whereas Embedded Development Kit (EDK) requires 2.06 s [2].

The MATLAB implementation is attractive for deploying AES into various applications such as image processing, video processing, etc.

VI. Conclusion

Advanced Encryption Standard algorithm is implemented in MATLAB for encrypting the plain text and decrypting the cipher text. The execution time is faster than the available published results. The highest decryption time is 88 ms, which is faster than the EDK implementation by Sambsiva Reddy et al [2]. This algorithm has wide range of applications and is tested in single board computers like Raspberry Pi, Arduino, Intel compute stick. Using Matlab, AES can be easily implemented for audio, image, video and signal encryption and decryption.

AUTHORS DESCRIPTION

1. D.LohitKumar is pursuing his M.Tech in the stream of Micro & Nano electronics from Madanapalle Institute of Technology and Sciences, Madanapalle. He completed his B.Tech in the stream of Electronics and communication, from Sreenivasa Institute of Technology And Management Studies, Chittoor. His areas of interest are Matlab, embedded systems, IC fabrication, and Micro &Nano electronics.

2. Dr.A.R.Reddy received his M.Sc from Osmania University and M.Tech in Electrical and Electronics and communication Engineering from IIT, Kharagpur during the year 1979 and his Ph.D degree from IIT, kharagpur during the year 1986.He worked as a senior scientist in R&D of ITI Ltd, Bangalore for about 24 years. He is currently working as a professor and head in the department of Electronics and Communication, Madanapalle Institute of Technology & Science. Madanapalle. His current research areas in Cryptography and its application to wireless systems and network security. He has published and presented papers in journals, international and national level conferences.

3.Dr. S A K Jilani is working as the Professor and project coordinator in the department of ECE, Madanapalle Institute of Technology and Sciences, Madanapalle. He has the teaching experience of over -twelve years. He also worked as an R&D Professi- onal earlier in Electronics Industry. He obtained his PhD In the year 2002 and also published more than 35 papers in different national and international Journals. He has also guided more than 50 M.Tech, M.Sc, MCA, B.Tech Projects. His areas of interest are Artificial Intelligence, Computer Visions, Digital Signal Processing and Embedded Systems.

REFERENCES

[1] Advanced Encryption Standard (AES), Federal Information, processing standards Publication 197, November26,2001.

[2]*M.SambasivaReddy,Mr.Y.AmbarBabu."Evaluation of Microblaze and implementation of AES Algorithm using Spartan 3-E.*"International Journal of Advanced Research In Electrical, Electronics and Instrumentation Engineering. ISSN (Print):2320-3765, ISSN(online):2278-8875.

[3] National Institute of Standards and Technology, "Federal Information Processing Standard Publication 197, the Advanced Encryption Standard(AES)," Nov. 2001.

[4] J. Daemen and V. Rijmen, *The block cipher Rijndael*, Smart Card research and Applications, LNCS 1820, Springer-Springer-Verlag, pp. 288-296

[5] L.Thulasimani, M.Madheswaran. "*Design And Implementation of Reconfigurable Rijndael Encryption Algorithms For Reconfigurable Mobile Terminals.*"(IJCSE) International Journal on Computer Science and Engineering.Vol. 02, No. 04, 2010, 1003-1011

[6]A. Lee, NIST Special Publication 800-21, *Guideline for Implementing Cryptography in the Federal Government,* National Institute of Standards and Technology, November 1999.

[7] Prashanti.G, Deepthi.S & Sandhya Rani.K. "*A Novel Approach for Data Encryption Standard algorithm*".International Journal Of Engineering and advancedTechnology.

[8] Das Debasis, Misra Rajiv. "*Programmable Cellular Automata Based Efficient Parallel AES Encryption Algorithm*". International Journal of Network Security & Its Applications(IJNSA) VOL.3 No.6, November 2011.

[9] N.Renuka Devi, G.Swathi,*"A high Throughput ASIC Implementation of configurable Advanced Encryption Standard(AES) Processor."*IJCA special issue on Network Security And Cryptography"NSC,2011.

[10] Kalpana Parsi, Singaraju Sudha."Data Security in Cloud Computing using RSA Algorithm" IJRCCT, ISSN 2278-5841, Vol 1, Issue 4, September 2012. pp. 145.

[11]M. Ravindra Babu and Dr. A. R. Reddy"Optimization Of Memory For AES Rijndeal Algorithm Implementation On Embedded System" International Journal of Engineering Research & Technology (IJERT). ISSN: : 2278-0181 Vol.1 Issue 7, September - 2012