Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 2, Issue 11, February 2019

AES Based Algorithm for Image Encryption and Decryption

Prof. Venkatesha G
Professor & HOD, Department of ECE, Brindavan College of Engineering, Bangalore, Karnataka, INDIA,
hodece@brindavancollege.com

Dr. Dinesh S
Associate Professor & HOD, Department of ISE, Brindavan College of Engineering, Bangalore, Karnataka, INDIA,
hodise@brindavancollege.com

Dr. Manjunath M
Assistant Professor, Department of ECE, Brindavan College of Engineering, Bangalore, Karnataka, INDIA,
manjuec043@brindavancollege.com

Abstract: Security in transmission storage of digital image has its important data and image communication. AES is a well-known cipher that has block cipher which has several advantages. AES algorithm used for text data and also suitable for image encryption and decryption to protects confidential image data from an unauthorized access. In this paper we are developing verilog code to implement 128 bits .AES for image encryption and decryption which is synthesized and simulated on FPGA family of Spartan-6 using Xilinx ISE 12.4 tool.

Keywords: AES, Image, Encrypt, Decrypt, FPGA, Cipher text, NIST

I. INTRODUCTION

With the development of Computer Network and Communication Technology, a great mass of data and information need to be exchanged by public communication networks. High efficiency and high safety of Data transmission become much more important. Cryptography plays an important role in the security of data. It enables us to store sensitive information or transmit it across insecure networks so that unauthorized persons cannot read it.

The exchange of digital data in cryptography results in different algorithm that can be classified into two cryptographic mechanisms[2].Symmetric key in which same key is used for encryption and decryption and asymmetric key in which different keys are used for encryption and decryptions. Symmetric key algorithms are much faster and easier to implement and generally requires less processing power when compared with asymmetric key algorithms.

In this cryptography, high efficiency and high safety of Data transmission become much more important. The appearance of DES (Data Encryption standard) solved many fields of information security problems. But, the security strength of DES is hard to adapt the new security needs by the development of password analysis level[1]. The National Institute of Standards and Technology (NIST) declared Rijndael algorithm as the Advanced Encryption Standard (AES) in October 2000[1].

Concept of Encryption and Decryption is shown in Figure 1.Original image on the left is encrypted using a key to generate encrypted image in the middle of the figure. When the same key is used decryption generates identical image. AES algorithm is not only for the text data, it can applied for the images, usually image processing deals with a image, which is composed of many image points[5].

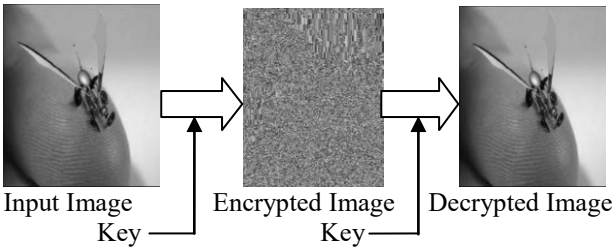

Fig 1. Concept of Encryption and Decryption

The image points, namely pixels, spatial co-ordinates that indicate the position of the points in the image and intensity values. The applications of the image processing have been commonly found in the Military communication, Forensics, Robotics, Intelligent systems etc. In this paper, we implement the AES algorithm which is an efficient scheme for both hardware and software implementation.

II. FUNCTIONAL BLOCK DIAGRAM

The functional block diagram of the system is shown in Fig 2. MATLAB is usually required to READ the image, which converts input image into its respective pixel values. Image which has to be secured is taken. The pixel values are given as input to AES-CORE with add-round key. AES undergo with encryption and decryption with its respective steps and the decrypted data, that is output can be seen on FPGA by writing Verilog code for the decrypted data. FPGA [Field Programmable Gate Array] supports for both hardware and software system. As we are using symmetric key which is a secret key,

same key should be used at both transmitter and receiver sides. When key matches at the receiver side the output can be on FPGA and these values are displayed on LCD. Keyboard is used to input the key that is symmetric key. The original image can be overviewed through MATLAB.

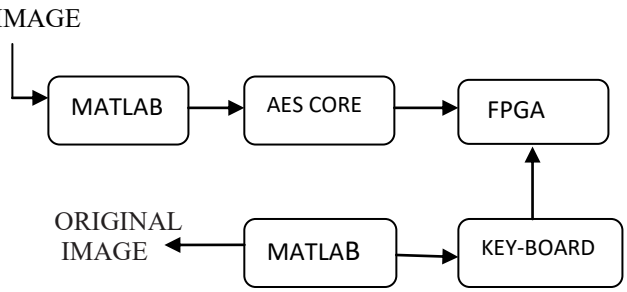

Fig 2. Functional block diagram

III. AES ALGORITHM

The algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. Cipher converts data to an unintelligible form called cipher text while Inverse Cipher converts data back into its original form called plaintext. Key Expansion generates a Key Schedule that is used in Cipher and Inverse Cipher procedure. Cipher and Inverse Cipher are composed of specific number of rounds.

It accepts block of Size 128, 192 or 256 bits. Independently the key length can be 128, 192 or 256 bits [2]. All encryption/decryption modes are done in 10, 12 or 14 round depends on the size of the block and the key length chosen as shown in the figure 2. AES merely allows a 128 bit data length that can be divided into four basic operation blocks [6]. These blocks operate on array of bytes and organized as a 4x4 matrix.

	Key Length	Block Size	Number of Rounds
AES 128	4	4	10
AES 192	6	4	12
AES 256	8	4	14

Table 1. Different AES Blocks

A. Encryption

AES algorithm begins with an Add round key stage followed by nine rounds with four stages and a tenth round of three stages which applies for both encryption and decryption algorithm[1][2][4].These rounds are governed by four stages:

- Substitute Bytes
- Shift rows
- Mix columns
- Add round key

The tenth round Mix columns stage is not included. The first nine rounds of the decryption algorithm are governed by the following four stages:

- Inverse Shift rows
- Inverse Substitute Bytes
- Add round key
- Inverse Mix columns

The Overall flow of the encryption and decryption algorithm of the AES algorithm is show in Fig 3.

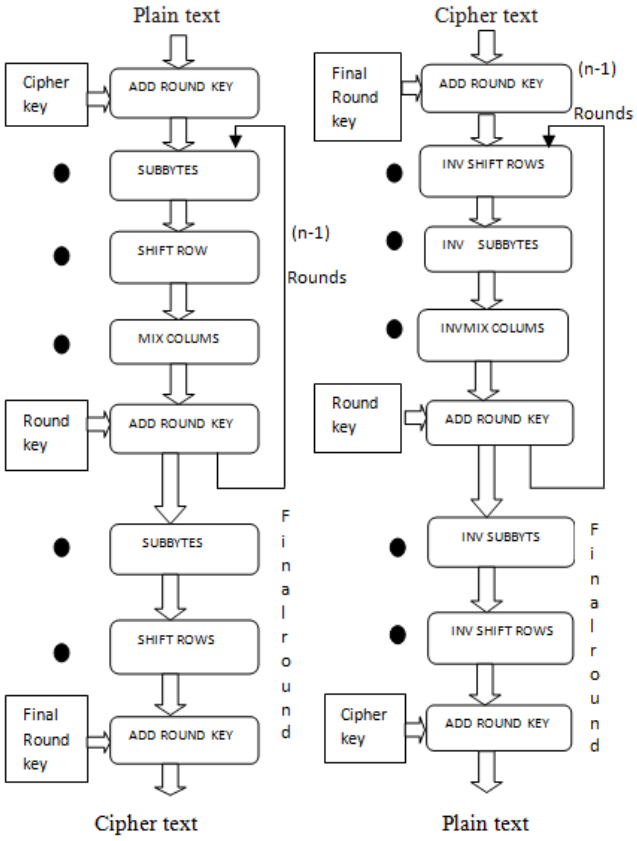

Fig 3. AES Design Flow

The four different transformations are described in detail below:

a) Sub Bytes Transformation: It is a non-linear substitution of bytes that operates independently on each byte of the State using a substitution table (S box) as shown in Figure 4. This S-box is constructed by using Galois Field-GF (2^8) with irreducible polynomial,

$$m(x) = x^8 + x^4 + x^3 + x + 1.$$

The element is mapped to itself. Then affine transformation is applied (over GF (2)).

b) Shift Rows Transformation: Cyclically shifts the rows of the State over different offsets. In this shift transformation the bytes in the last three rows of the state are cyclically shifted over 1, 2, 3 bytes to the

left, respectively. First row is not shifted as shown in the figure 6. The operation is almost the same in the decryption process except for the fact that the shifting offsets have different values.

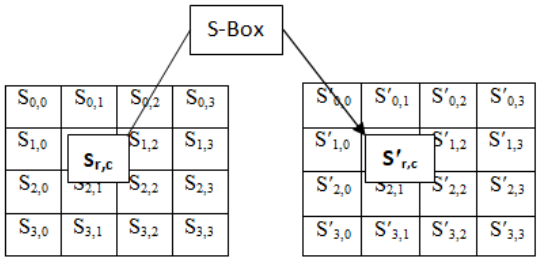

Fig 4. Sub-byte operation

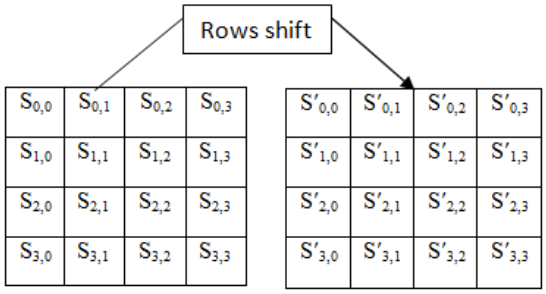

Fig 5. Shift rows Operation

c) *Mix Columns Transformation:* This transformation operates on the State column-by-column, treating each column as a four-term polynomial[6]. The columns are considered as polynomials over GF (2^8) and multiplied by modulo $x^4 + 1$ with a fixed polynomial equation,

$$a(x) = \{03\}\ x^3 + \{01\}\ x^2 + \{01\}\ x + \{02\}$$

The result is the corresponding column of the output state is as shown in Figure 6.

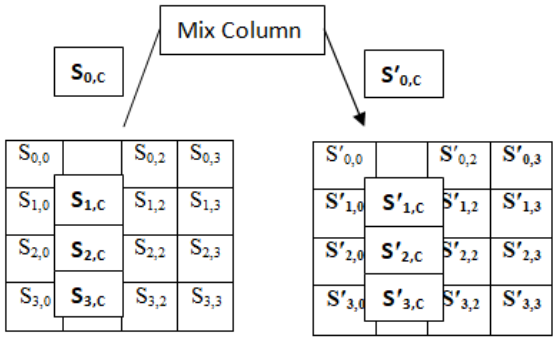

Fig 6. Mix Column Operation

d) *Add Round Key Transformation:* The Add Round Key operation is as shown in Figure 8, which is a simple XOR operation between the State and the Round Key. The Round Key is derived from the Cipher key by means of key schedule process. The State and Round Key are of the same size and to obtain the next State an XOR operation is done per element:

$$b(i, j) = a(i, j) \oplus k(i, j)$$

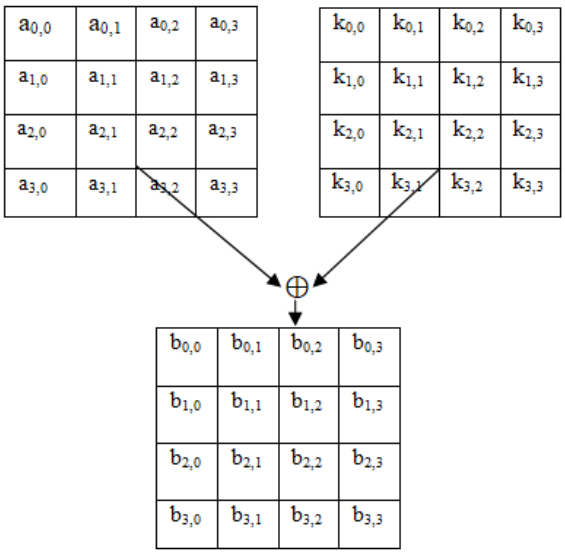

Fig 7. Add Round Key Operation

Key Expansion: Each round key is a 4-word (128-bit) array generated as a product of the previous round key, a constant that changes each round [9], and a series of S-Box lookups for each 32-bit word of the key.

B. *Decryption*

For decryption, the same process occurs simply in reverse order – taking the 128-bit block of cipher text and converting it to plaintext by the application of the inverse of the four operations. AddRoundKey is the same for both encryption and decryption [3][7]. However the three other functions have inverses used in the decryption process: Inverse Sub Bytes, Inverse Shift Rows, and Inverse Mix Columns.

This process is direct inverse of the Encryption process. All the transformations applied in Encryption process are inversely applied to this process.

IV. IMPLEMENTATION

A. *AES Encryption*

The AES Encryption block is as shown in Figure 8, the encryption parameters are the input plaintext, the key size and the output cipher text. First, we have to map the 16 byte input plaintext in the correct order to the 4x4 byte state, calculate the number of rounds based on the key Size and expand the key using our key schedule. At round one plaintext and key is XORed, the remaining nine rounds which has to apply all four operations: Substitute Bytes, Shift rows, Mix columns, Add round key. The tenth round Mix columns stage is not included and round key was generated during each iteration [7]. Here simple XOR of each byte of the key with the respective byte of the state is done to get cipher text of the Encryption algorithm [10].

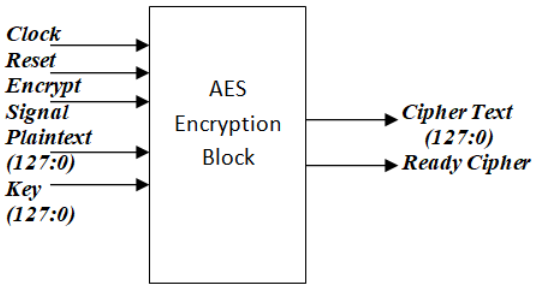

Fig 8. AES Encryption Block

B. AES Decryption

AES Decryption, the same encryption process occurs simply in reverse order. The decryption block is as shown in Figure 9, the encryption parameters are the input cipher text, the key and the output plaintext should be same as encryption input [8]. In decryption the key schedule remains the same; the only operations we need to implement are the Inverse Sub Bytes, Shift Rows and Mix Columns, while Add Round Key stays the same.

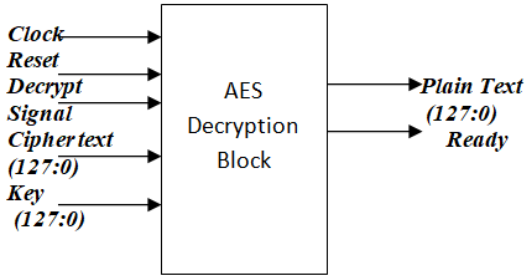

Fig 9. AES Decryption Block

V. SIMULATION RESULTS

The waveforms generated for the Image Encryption and Decryption Process is shown in the figure below.

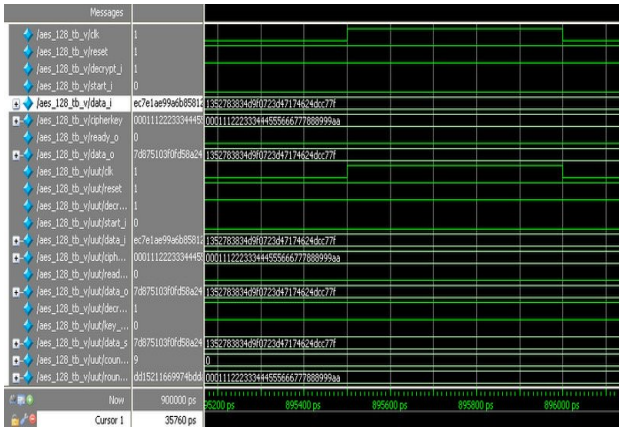

Fig 10. Simulation Waveform

Encryption Process (128 bit)

Plain Text: 00112233445566778899aabbccddeeff.

Key: 000102030405060708090a0b0c0d0e0f.

Cipher Text: 69c4e0d86a7b0430d8cdb78070b4c55a.

Decryption Process (128 bit)

Cipher Text: 69c4e0d86a7b0430d8cdb78070b4c55a.

Key: 000102030405060708090a0b0c0d0e0f.

Plain Text: 00112233445566778899aabbccddeeff.

VI. CONCLUSION

A Successful implementation of AES algorithm is one of the best encryption and decryption standard available in market. In this paper, Image Encryption and Decryption using AES is designed and implemented to protect the confidential image data from an unauthorized access. It helps to explore the path to implement such an algorithm using Verilog code that is synthesized and simulated using the ISE 12.4 in Xilinx Family Spartan-6 The Maximum Frequency achieved from the design is 165.462MHz and the throughput reaches the value of 252.132Mbit/sec for Image Encryption and Decryption.

REFERENCES

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

[2] William Stalling, Cryptography and Network Security: Principles and Practices, 4th ed. Prentice Hall, 2006.

[3] Xinmiao Zhang, Keshab K. Parhi, Fellow, "High-Speed VLSI Architectures for the AES Algorithm" IEEE Transactions on vlsi systems, vol. 12, no. 9, pp.957-966, September 2004.

[4] Abdulkarim Amer Shtewi, Bahaa Eldin M. Hasan, Abd El Fatah .A. Hegazy "An Efficient Modified Advanced Encryption Standard (MAES) Adapted for Image Cryptosystems" IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.2,pp.226-232 February 2010.

[5] Network Security G.Ramesh, Prof. Dr. R. "Umarani: A novel fast encryption algorithm for data security in local networks"978 -1-4244 -7770 -8/10/$26.00 ©2010 IEEE.

[6] Top Down Implementation of Pipelined AES Cipher and its Verification with FPGA-based Simulation Accelerator. Jae-Gon Lee,Woong Hwangbo, Seonpil Kim and Chong-Min Kyung.Department of EEEC.Emai:{jglee,woonghb,spkim,kyung}@vslab.kaist.ac.kr.

[7] "Design and Implementation of area optimized AES algorithm on reconfigurable FPGA" Ahmed Rady Ahmeed_rady@yahoo.com, Ehab EL Sehely elsehely@yahoo.com, A.M. EL Hennawy a.hennawy@marsem.com.

[8] "Comparison of various strategies of implementation of the algorithm of encryption AES on FPGA" Oscar Perez, Yves Berviller, Camel Tanougast, Serge Weber oscar.perez*ien.uhp-nancy.fr .

[9] P.Karthigaikumar, Soumiya Rasheed "Simulation of Image Encryption using AES Algorithm" IJCA Special Issue on "Computational Science - New Dimensions & Perspectives" NCCSE, pp166-172, 2011.

[10] "The AES Application in Image Using Different Operation Modes" Chi-Wu Huang1, Che-Hao Chiang2, Chien-Lun Yen2, Yi-Cheng Chen1, Kuo-Huang Chang2 and Chi-Jeng Chang2.Email: t07022@ntnu.edu.tw.