ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

FPGA BASED AES IMPLEMENTATION FOR DATA SECURITY

M. Pavithra,
M.E. VLSI Design student
Department of Electronics and Communication Engineering
TRP Engineering College , Irrungalur ,Trichy-621105
Email:mpavithra51093@gmail.com

R. ArunaJayashree
Assistant Professor
Department of Electronics and Communication Engineering
TRP Engineering College, Irrungalur,Trichy-621105

Abstract **—The increasing need for protecting data communication in computer networks has led to development of several cryptography algorithms. The Advanced Encryption Standard (AES) is a computer security standard issued by the National Institute of Standards and Technology (NIST) intended for protecting electronic data..The AES cryptography algorithm can be used to encrypt/decrypt blocks of 128 bits and is capable of using cipher keys of 128, 196 or 256 bits wide (AES128, AES196, and AES256).Its specification is defined in Federal Information Processing Standards (FIPS) Publication197.In order to achieve higher performance in today's heavily loaded communication networks, utilization of hardware accelerators for cryptography algorithms is more efficient. This paper investigates a novel attack vector against cryptography realized on FPGAs, which poses a serious threat to real-world applications. A proposed FPGA-based implementation of the Advanced Encryption Standard (AES) algorithm is presented in this paper. The design uses an iterative looping approach with block and key size of 128 bits, lookup table implementation of S-box. The design has been coded by Very high speed integrated circuit Hardware Descriptive Language. All the results are synthesized and simulated using Xilinx ISE and ModelSim software respectively. This gives low complexity architecture and easily achieves low latency as well as high throughput.**

Index Terms **— Advanced Encryption Standard (AES), Field Programmable Gate Arrays (FPGAs), Hardware Security, Encryption , Decryption, Block cipher and VHDL.**

I. INTRODUCTION

FIELD Programmable Gate Arrays (FPGAs) play an Important role in the field of embedded systems. They are used in a wide spectrum of applications, e.g., computer networks, data centers, automation, signal processing, and the automotive industry [1].Many of these applications are security sensitive and use FPGAs for cryptographic operations such as random number generation, key establishment, digital signatures as well as encryption. Data encryption is achieved by following a systematic algorithm called encryption algorithm. An encryption algorithm provides Confidentiality, Authentication, Integrity and Non-repudiation. Confidentiality is the requirement that information is kept secret from people who are not authorized to access it. Authentication is the certainty that the message indeed originates from the purported sender [5].Integrity is the requirement that information is unaltered and complete, or, that information "is modified only by those users who have the right to do so." Nonrepudiation means that the sender or receiver of a message cannot deny having sent or received the message.

For a long time, the Data Encryption Standard (DES) was considered as a standard for the symmetric key encryption. DES has a key length of 56 bits. However, this key length is currently considered small and can easily be broken. For this reason, the National Institute of Standards and Technology (NIST) opened a formal call for algorithms in September 1997.In August 2000, NIST selected five algorithms: Mars, RC6, Rijndael, Serpent and Twofish as the final competitors. These algorithms were subject to further analysis prior to the selection of the best algorithm for the AES [12].

Finally, on October 2, 2000, NIST announced that the Rijndael algorithm was the winner. Field Programmable Gate Arrays are hardware devices whose function is not fixed which can be programmed in system [2]. The potential advantage of encryption algorithm implemented in FPGAs includes: Algorithm agility- this term refers to the

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

switching of cryptographic algorithm during operation. Algorithm upload- it is perceivable that fielded devices upgraded with new encryption algorithm which did not exist at design time.
Algorithm modification - There are applications which require modification of a standardized algorithm. Architecture efficiency- With FPGAs it is possible to design and optimize architecture for specific parameter set. Throughput- Although typically slower than ASIC implementation, FPGA have potential of running substantially faster than software implementations. Cost efficiency- Time and cost for developing an FPGA implementation of a given algorithm are much lower than for an ASIC implementation.
In cryptography, the AES is also known as Rijndael. AES has a fixed block size of 128 bits and a key size of 128,192 or 256 bits. This paper deals with an FPGA implementation of an AES encryptor/decryptor using an iterative looping approach with block and key size of 128 bits. This method gives very low complexity architecture and is easily [4] operated to achieve low latency as well as high throughput. In this paper organized following sections like, section 2 describes AES , section 3 tellsdetail about proposed work of this paper and section 4 describes simulation result. Finally, section 5 tells the conclusion of this paper.

II. ADVANCED ENCRYPTION STANDARD

The AES is a cryptographic algorithm that is used to encrypt (encipher), and decrypt, (decipher), information. Key Expansion generates a Key Schedule that is used in Cipher and Inverse Cipher procedures. Cipher and Inverse Cipher are composed of specific number of rounds (Table 1). For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key length[3].For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is composed of four different byte oriented transformations.

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

Thereby, the SubBytes step processes sixteen intermediate bytes by using a fixed S-box. In addition to that, the key schedule step also needs to process four S-box instances [15]. The basic processing unit for the AES algorithm is a byte. As a result, the plaintext, ciphertext and the cipher key are arranged and processed as arrays of bytes.

Type	Block Size Nb words	Key Length Nk words	Number of Rounds Nr
AES-128	4	4	10
AES-192	4	6	12
AES-256	4	8	14

Table.1. Comparison of block size, key length and number of rounds of AES keys

AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. However, AES is quite different from DES in a number of ways [8]. The algorithm Rijndael allows for a variety of block and key sizes and not just the 64 and 56 bits of DES' block and key size. The basic structure of AES shown in (Fig.1).The AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits. Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES-256 respectively.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

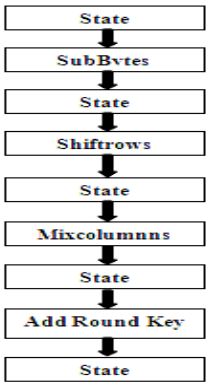

Fig.1 Basic structure of AES

The input is a single 128 bit block both for decryption and encryption and is known as the **in** matrix. This block is copied into a **state** array which is modified at each stage of the algorithm and then copied to an output matrix [7]. Both the plaintext and key are depicted as a 128 bit square matrix of bytes. This key is then expanded into an array of key schedule words. The over all process of the aes algorithm shown in (Fig.2).All the AES algorithm operations are performed on a two dimensional 4x4 array of bytes which is called the ***State***.

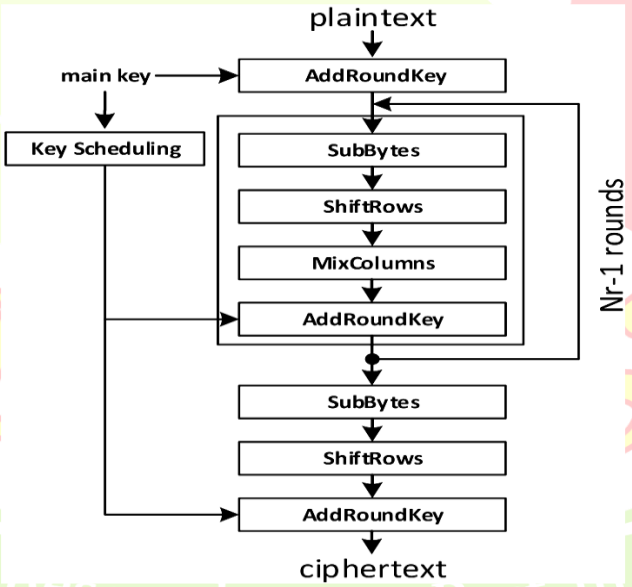

Fig.2. Overview of the AES encryption algorithm.

III. PROPOSED WORK

Encryption is the process of converting the plain text into a format which is not easily readable and is called as cipher. The cipher is got by doing a series of mathematical operations iteratively [6]. The AES algorithm can be implemented in both hardware and software. The software implementation of AES algorithm is a slow process when compared with hardware process. The AES algorithm is a symmetric block cipher that can encrypt and decrypt information [10]. Encryption converts data to an unintelligible form called cipher-text. Encryption of the cipher-text converts the data back into its original form, which is called plain-text. AES Encryption/Decryption process shown in (Fig.3).

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

A.AES encryption

In this paper, we use the key length of 128 bits (AES-128) as a model for general explanation. So AES algorithm operates on a 128-bit block of data and executed number of loop times.The key length is 128, 192 or 256 bits in length respectively. The first and last rounds differ from other rounds in that there is an additional AddRoundKey transformation at the beginning of the first round and no MixCoulmns transformation is performed in the last round.

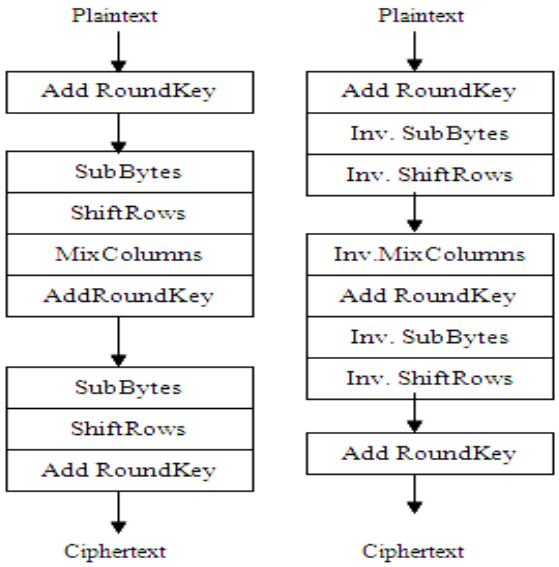

Fig.3. AES Encryption/Decryption process

1 .SubBytes Transformation:

The SubBytes transformation is a non-linear byte substitution, operating on each of the state bytes independently. The SubBytes transformation is done using a once-precalculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values[11]. This approach has the significant advantage of performing the S-box computation in a single clock cycle, thus reducing the latency and avoids complexity of hardware
implementation.

2. ShiftRows Transformation:

In ShiftRows transformation, the rows of the state are cyclically left shifted over different offsets. Row 0 is not shifted; row 1 is shifted one byte to the left; row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left.

3. *MixColumns Transformation:*

In MixColumns transformation, the columns of the state are considered.
as polynomials over GF (2^8) and multiplied by modulo $x^4 + 1$ with a fixed polynomial c(x), given by:
$c(x)=\{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$.

4. AddRoundKey Transformation:

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

In the AddRoundKey transformation, a Round Key is added to the State - resulted from the operation of the MixColumns transformation - by a simple bitwise XOR operation. The RoundKey of each round is derived from the main key using the KeyExpansion algorithm. The encryption/ decryption algorithm needs eleven 128-bitRoundKey [9], which are denoted RoundKey[0] - RoundKey[10].

B. AES decryption

Decryption is a reverse of encryption which inverse round transformations to compute the original plaintext of an encrypted cipher-text in reverse order. key function of decryption is same from encryption The round transformation of decryption uses the functions are following AddRoundKey, InvMixColumns, InvShiftRows, and InvSubBytes successively.

1.AddRoundKey:

AddRoundKey is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order.

2. InvShiftRows Transformation:

InvShiftRows exactly functions the same as ShiftRows, only in the opposite direction. The first row is not shifted, while the second, third and fourth rows are shifted right by one, two and three bytes respectively.

3. InvSubBytes transformation:

InvSubBytes transformation is done to precalculated substitution table called InvS-box. That the InvS-box table contains 256 numbers (from 0 to 255) and their corresponding values.

4. InvMixColumns Transformation:

The InvMixColumns transformation is done by polynomials of degree less than 4 over GF(2_8), which coefficients are the elements in the columns of the state, are multiplied modulo ($x_4 + 1$) by a fixed polynomial $d(x)= \{0B\}x_3 + \{0D\}x_2 + \{09\}x + \{0E\}$, where $\{0B\}$, $\{0D\}$;$\{09\}$, $\{0E\}$ denote hexadecimal values.
In that the encryption ,decryption algorithm is presented in this paper. We use this algorithm implement [14] the data security process and obtain the request data in secure manner.

Algorithm for AES encryption:

```
byte state[4,Nb]
state = in
AddRoundKey(state, keySchedule[0, Nb-1])
for round = 1 step 1 to Nr–1
{
SubBytes(state)
ShiftRows(state)
MixColumns(state)
AddRoundKey(state,keySchedule[round*Nb, (round+1)*Nb-1])
}
SubBytes(state)
```

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

```
ShiftRows(state)
AddRoundKey(state,keySchedule[Nr*Nb,(Nr+1)*Nb-1])
out = state
```

Algorithm for AES decryption:

```
byte state[4,Nb]
state = in
AddRoundKey(state, keySchedule[Nr*Nb, (Nr+1)*Nb-1])
for round = Nr-1 step -1 downto 1
 {
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, keySchedule[round*Nb,(round+1)*Nb-1])
InvMixColumns(state)
}
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, keySchedule[0, Nb-1])
out = state
```

IV. SIMULATION RESULT

The design of AES algorithm is coded by VHDL language. Simulation and synthesis of AES algorithm is done on ModelSim software and Xilinx ISE software respectively. We implemented the AES Encryption/Decryption module on a Xilinx XC3S100E Spartan-3E FPGA kit.

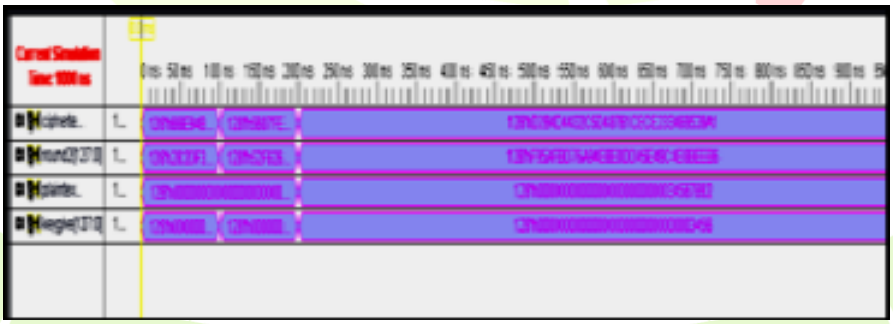

Fig.5 Data encryption result

Simulation result of data encryption and decryption presented in this paper (Fig.5 & Fig.6). The design flow of this project is given below. From design specification, design will be coded using Very High Speed Integrated Circuit Hardware Descriptive Language Simulation and verification will be done on ModelSim software. Results are then synthesized on Xilinx ISE. Generated Bitstream file will need to program the FPGA.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

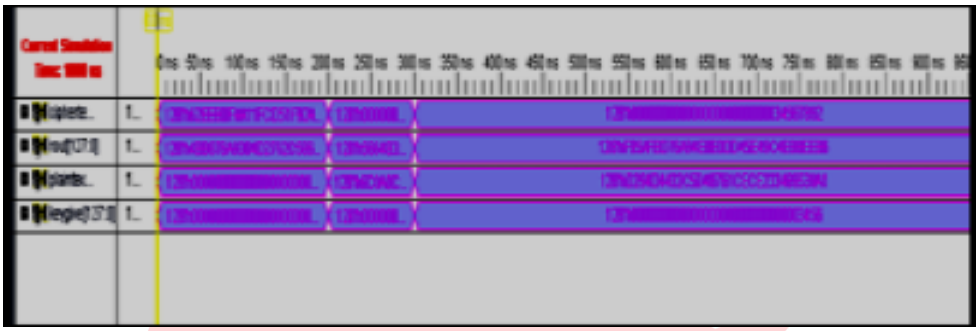

Fig.6 Data decryption result

V. CONCLUSION

The AES algorithm can be efficiently implemented using software. The Software implementations cost the smallest
resources, but they offer a limited physical security and the slowest process. Besides, growing requirements for high speed, high volume secure communications combined with physical security, hardware implementation of cryptography takes place. We are going to design FPGA based hardware implementation of AES algorithm. The design is coded using VHDL language. The design is simulated on ModelSim software and synthesize on Xilinx ISE software. In future the image also transferred in secure manner using AES algorithm.

REFERANCES

[1] Pawel Swierczynski , Mar Fyrbiak , Philipp Koppe and Paar Fellow " FPGA Trojans Through Detecting and Weaking of Cryptographic Primitives" in 2015 IEEE.

[2] Yang Jun Ding Jun Li Na Guo Yixiong " FPGA Based Design and Implementation of Reduced AES Algorithm" in 2010 IEEE 3972-0/10.

[3] Hoang Trang and Nguyen Van" An Efficient FPGA Implementation Of Advanced Encryption Standard Algorithm " in 2012 IEEE 309-5.

[4] Wang Wei, CHAN Jie & XU Fei,China " An Implementation of AES Algorithm Based on FPGA" in 2012 IEEE.

[5] V. Rijmen, "Efficient Implementation of the Rijndael S-box" , Available at http://www.esat.kuleuven.ac.be/rijndael/-rijmen.

[6] Abidalrahman Moh'd, YaserJararweh, Lo'ai Tawalbeh, " AES- 512 512 - Bit Advanced Encryption Standard Algorithm Design and Evaluation ",in 2011 IEEE pp 29-32.

[7] Gurmail Singh, Rajesh Mehra " FPGA Based High Speed and Area Efficient AES Encryption for Data Security " in 2011 IEEE

[8] Refik Sever, A. Neslin smailoglu, Yusuf C. Tekmen " A High Speed fpga Implementation of the Rijndael Algorithm" IEEE
2004.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

[9] Atul M. Borker, R. V Kshirsager. M. V. Vyawahare " FPG Implementation of AES Algorithm" IEEE 2011.

[10] Sumanth Kumar Reddy S, R. Sakthivel, P. Prasanth " VLSI Implementation of AES Crypto Proceesor for High Throughput"IEEE 2011.

[11] A. Moradi , A. Barenghi , T . Kasper and C . Peer " On the Vulnearablity of FPGA Bitstream Encryption Against Power Analysis from Xilinx Virtex-II FPGAs" in 2011 IEEE.

[12] "Advanced Encryption Standard (AES)", NIST FIPS Standard 197, 2001.[online]. Available: http//csrc.nist.gov/publications/fips/ Fips 197/fips-197.pdf

[13] H. Johnson, S. Chow ,P. A. Eisen and P.C Van Oorshot, "Whit Box Cryptography and AES Implementation " Springer in 2002.

[14] R. Chakraborty, I Ssha , A. Palchaudhuri, and G.Nalik " Hardware Trojen Insertion by Direct Modification of FPGA Bitstream" IEEE 2013.

[15] C.S Jutla "Encryption Modes with Almost Free Message Integrity" Springer in 2001.