International Journal of Emerging Engineering Research and Technology
Volume 3, Issue 4, April 2015, PP 104-108
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Implementation of AES-256 Encryption Algorithm on FPGA

[1]Dasari. Subbarao, [2]B. Swapnakumari

[1]Associate Professor, Siddhartha Institute of Engg & Technology, Ibrahimpatanam
[2]Assistant Professor, Siddhartha Institute of Engg & Technology, Ibrahimpatanam

ABSTRACT

The Advanced Encryption Standard can be programmed in software or built with pure hardware. However Field Programmable Gate Arrays (FPGAs) offer a quicker, more customizable solution. In the implementation of this AES-256 algorithm has a plaintext of 128 bits and key of 256 bits size. The number of rounds of operations in AES- 256 is 14. The key generation process of AES 256 is different from other AES algorithms. This research investigates the AES algorithm with regard to FPGA and the Very High Speed Integrated Circuit Hardware Description language (VHDL). Xilinx 9.2i software is used for simulation and optimization of the synthesizable VHDL code. All the transformations of both Encryption and Decryption are simulated using an iterative design approach in order to minimize the hardware consumption. Xilinx XC3S500 device of Spartan Family is used for hardware evaluation.

Keywords: AES algorithm (encryption, decryption), key expansion, hardware implementation.

INTRODUCTION

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 urgency for secure ex-change of digital data resulted in large quantities of different encryption algorithms which can be classified into two groups: asymmetric encryption algorithms (with public key algorithms) and symmetric encryption algorithms (with private key algorithms). Symmetric key algorithms are in general much faster to execute electronically than asymmetric key algorithms.

The Advanced Encryption Standard, in the following referenced as AES, is the winner of the contest, held in 1997 by the US Government, after the **Data Encryption Standard** was found too weak because of its small key size and the technological advancements in processor power. Fifteen candidates were accepted in 1998 and based on public comments the pool was reduced to five finalists in 1999. In October 2000, one of these five algorithms was selected as the forthcoming standard: a slightly modified version of the Rijndael.

The National Institute of Standards and Technology, (NIST), solicited proposals for the Advanced Encryption Standard, (AES). The AES is a Federal Information Processing Standard, (FIPS), which is a cryptographic algorithm that is used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt, (encipher), and decrypt, (decipher), information. Encryption converts data to an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of bits. We are implementing AES-256 algorithm with key length 256 bits for acquiring higher security for increasing the confidentiality, instead of AES-128 and AES-192 because as they are having less key length size when compared with AES-256. The need for privacy has become a high priority for both governments and civilians desiring protection from signal and data interception. Widespread use of personal communications devices has only increased demand for a level of security on previously insecure communications. Both DES (Data Encryption Standard) and AES are defined as symmetric key block ciphers, with the main difference being the bit length of the key (56 bit for DES). These symmetric-key encryption schemes use the same key for both the sender and receiver, and as a result eliminate the need for the verification server needed in public keying. Symmetric keying lends itself to work independently of an open network and in turn a higher level of system interoperability.

Address for correspondence

subbaraod15@gmail.com

THE AES ALGORITHM

The AES-256 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

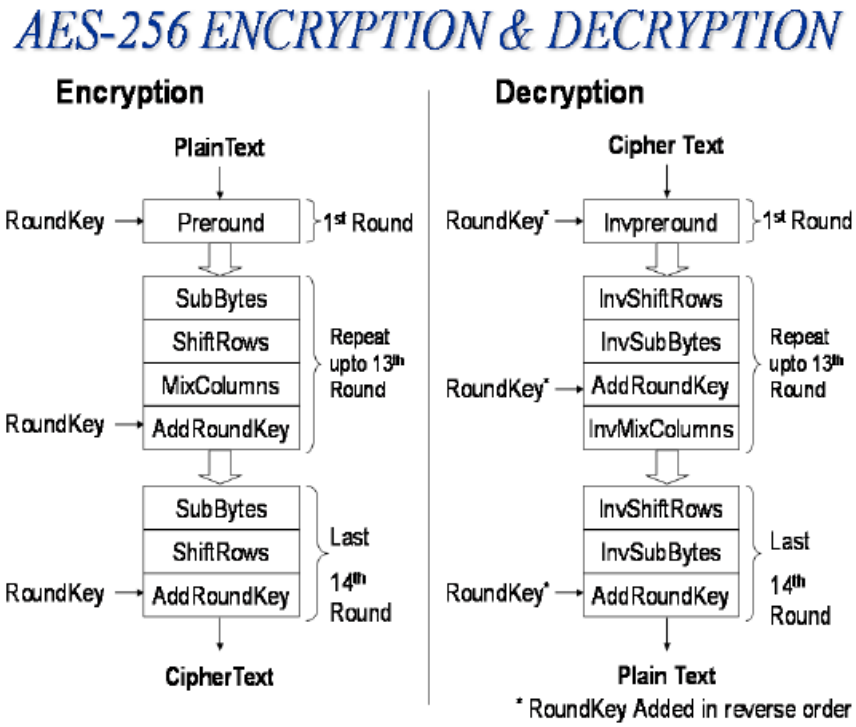
For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is composed of four different byte-oriented transformations:

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

The Cipher transformations can be inverted and then implemented in reverse order to produce a straightforward Inverse Cipher for the AES algorithm. The individual transformations used in the Inverse Cipher.

1) Inverse Shift Rows
2) Inverse Sub Bytes
3) Inverse Mix Columns
4) Add Round Key

The AES inverse cipher core consists of a key expansion module, a key reversal buffer, an initial permutation module, a round permutation module and a final permutation module. The key reversal buffer first store keys for all rounds and the presents them in reverse order to the rounds. The round permutation module will loop maternally to perform 14 iterations (for 256 bit keys).



IMPLEMENTATION OF AES 256

In the Encryption process we have a plaintext of 128 bits and key of 256 bits size. The number of rounds in AES 256 is 14. The first round consists of all the five operation like Preround operation, sub byte, shift rows, mix columns and Add round key operations. From 2nd round to 13th round have four operations sub byte, shift rows, mix columns and Add round key operations. And the last 14th round consists of three operations sub byte, shift rows and Add round key operations.

Description of the AES-256 Encryption algorithm

1. Key Expansion round keys are derived from the cipher key using Rijndael's key schedule.
2. Initial Round
    1. Add Round Key each byte of the state is combined with the round key using bitwise xor.
    2. Sub Byte a non-linear substitution step where each byte is replaced with another according to a lookup table.
    3. Shift Rows a transposition step where each row of the state is shifted cyclically a certain number of steps.
    4. Mix Columns is a mixing operation which operates on the columns of the state, combining the four bytes in each column.
    5. Add Round Key
3. Rounds(2nd to 13th)
    1. Sub Bytes
    2. Shift Rows
    3. Mix Columns
    4. Add Round Key.
4. Final Round (no Mix Columns)
    1. Sub Bytes
    2. Shift Rows
    3. Add Round Key.

In AES 256 the process of generating the key is each round key is a 256-bit array generated as follows,

Input key of 256 bit is divided into eight parts of 32 bits as columns in a matrix4*8. Last column is taken and given as input to S box. The output of S box is given shift rows operation. The above output MSB side 8 bits are xored with the round constant(i.e. round constant value is different for different rounds). The above output is xored with 0^{th} column of input key .The above output is taken as 0^{th} column of the generated new key. 0^{th} column of new key is xored with 1^{st} column of input key gives 1^{st} column of new key. 1^{st} column of new key is xored with 2^{st} column of input key gives 2^{nd} column of new key. 2^{nd} column of new key is xored with 3^{rd} column of input key gives 3^{rd} column of new key. The above obtained 3^{rd} column of new key is given to S box. The output of S box is xored with 4^{th} column of input key which gives 4^{th} column of new key. 4^{th} column of new key is xored with 5^{th} column of input key which gives 5^{th} column of new key. 5^{th} column of new key is xored with 6^{th} column of input key which gives 6^{th} column of new key. 6^{th} column of new key is xored with 7^{th} column of input key which gives 7^{th} column of new key. In this way we generate new keys of 256 bits in AES 256 algorithm by attaching the eight obtained columns of new key.

In the first round this key of 256 bits is divided into two parts each of 128 bits size and these keys of 128 bits are used one in the preround operation (i.e. xor operation between plaintext and key) and other is used in Add Round key operation. At the end of round function there will be 128bit output and 256 bit key output obtained from key generation process.

In the second round as we don't have preround operation so the round output of 1^{st} round is applied as input to sub byte and the remaining operations are same as round 1. This process of round operation is repeated up to 13^{th} round operation. And the last round is similar to previous round the only change is it doesn't have mix columns operation.

AES Decryption process is the inverse process of encryption AES. The output of the encryption process i.e. cipher text is given as input to decryption. The same input key of 256 bits is used as another input. The input key is given to the key generation module to generate new keys as we do in the encryption process. The output keys generated are given as inputs to inverse mix columns which give keys for the fourteen rounds of decryption.

Now in the first round all the five operation like Preround operation, sub byte, shift rows, mix columns and Add round key operations. From 2nd round to 13th round have four operations sub byte, shift rows, mix columns and Add round key operations. And the last 14th round consists of three operations sub byte, shift rows and Add round key operations. As in the encryption process the output of first round is taken as input to the next round up to the final fourteenth round and the output of fourteenth round is taken as final output.

COMPARISON WITH AES-128

In the implementation of AES128-bit algorithm as the name itself says that it has a key of 128 bit size, plain text of 128 bits and the output cipher text is of 128 bits size.

In the AES 128 we have 10 rounds of operations to be carried out. The first round consists of all the five operation like Preround operation, sub byte, shift rows, mix columns and Add round key operations. From 2nd round to 9th round have four operations sub byte, shift rows, mix columns and Add round key operations. And the last 10th round consists of three operations sub byte, shift rows and Add round key operations.

Algorithm	Key length, N_k	Block size, N_b	No of rounds, $N_r = N_k + 6$
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

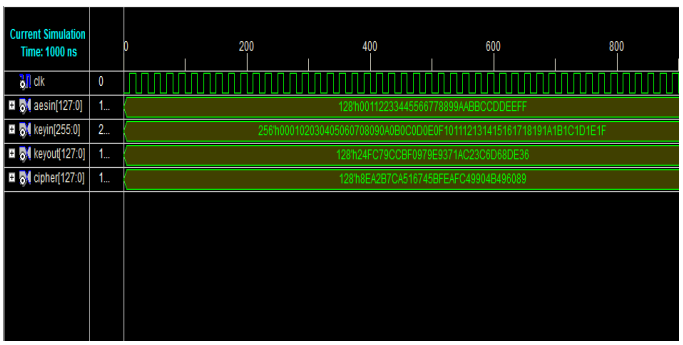
By comparing with the other versions of the AES algorithm we came to know that AES-256 provide high security to protect classified information.

The National Security Agency (NSA) reviewed all the AES finalists, including Rijndael, and stated that all of them were secure enough for U.S. Government non-classified data. In June 2003, the U.S. Government announced that AES could be used to protect classified information.

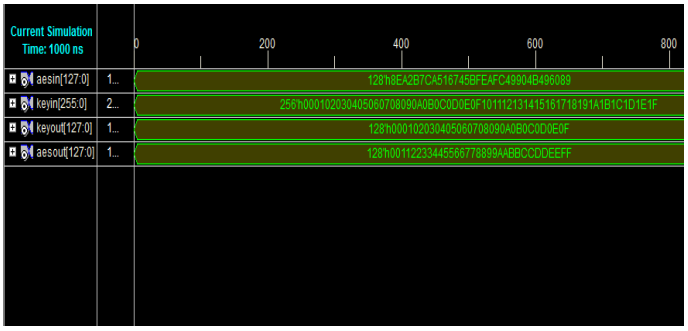
The design and strength of all key lengths of the AES algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the SECRET level. TOP SECRET information will require use of either the 256 key lengths. The implementation of AES in products intended to protect national security systems and/or information must be reviewed and certified by NSA prior to their acquisition and use.

SIMULATION RESULTS

Encryption Process



Decryption Process



CONCLUSION

The Advanced Encryption Standard algorithm is an iterative private key symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys with lengths of 128, 192, and 256 bits. An efficient FPGA implementation of 128 bit block and 256 bit key AES cryptosystem has been presented in this paper. Optimized and Synthesizable VHDL code is developed for the implementation of both 128 bit data encryption and decryption process & description is verified using ISE 9.2i functional simulator from Xilinx. All the transformations of algorithm are simulated using an iterative design approach in order to minimize the hardware consumption. Each program is tested with some of the sample vectors provided by NIST. The throughput reaches the value of 352Mbit/sec for both encryption and decryption process with Device XC3S500 of Xilinx Spartan Family.

REFERENCES

[1] Marko Mali, Franc Novak and Anton Biasizzo "Hardware Implementation of AES Algorithm" – Journal of ELECTRICAL ENGINEERING, Vol. 56, No. 9-10, 2005, 265-269.

[2] Behrouz A. Forouzan and Debdeep Mukhopadhyay "Cryptography and Network Security" (2nd edition).

[3] FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[4] L.Thulasimani,"A Single Chip Design and Implementation of AES-128/192/256 Encryption Algorithms"- International Journal of Engineering Science and Technology, Vol. 2(5), 2010, 1052-1059.

[5] Nation Institute of Standards and Technology (NIST), Data Encryption Standard (DES), National Technical Information Service, Sprinfgield, VA 22161, Oct. 1999.

[6] J. Daemen and V. Rijmen, "AES Proposal: Rijndael", AES Algorithm Submission, September 3, 1999.

[7] J. Nechvatal et. al., Report on the development of Advanced Encryption Standard, NIST publication, Oct 2, 2000.

[8] FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[9] K. Gaj and P. Chodowiec, Comparison of the hardware performance of the AES candidates using reconfigurable hardware, in The Third AES Candidates Conference, printed by the National Institute of Standards and Technology.