ISSN: 2456-043X

International Journal of Engineering Science and Generic Research (IJESAR)
Available Online at www.ijesar.in
Volume 3; Issue 5; September-October; 2017; Page No. 21-24

Effective Implementation of Dual Key Based AES Encryption with Key Based S-Box Generation

[1] Manohar Nagallapati, [2] K. Naveen Kumar Raju

[1] PG Scholar, Vaagdevi Institute of Technology and Science, Kadapa (dist), AP. India.

[2] Assistant Professor, Vaagdevi Institute of Technology and Science, Kadapa (dist), AP. India.

Abstract

Network security is an emerging domain of communication. Cryptography plays an important role in providing a secure network for communication. The most secure block cipher today is Rijndael cipher also known as AES. But with advanced research happening in the field of cryptography, the conventional scheme of AES is vulnerable for cryptanalysis. Hence a lot of changes have been proposed on this algorithm. Static S-Boxes are implemented using look up tables which will never vary with the input text or input key. This consumes a lot of Memory for the storage of look up table. Also this technique makes reverse engineering very simple for the purpose of cryptanalysis. Thus it is essential to generate S-Bytes at run time. It is beneficial if the S-byte generated during run time varies with the input key. Another weakness of AES is that it works with a single key. The confidentiality of the key determines the security of the algorithm. In this paper, a new scheme of AES involving generation of Key based S-Boxes and dual key AES is proposed. This overcomes the vulnerability of static S-Boxes and also single key encryption scheme. In this paper, the architecture of the algorithm for optimal FPGA implementation is also proposed.

Keywords: *AES, S-Box, Dual key, FPGA*

INTRODUCTION

Encryption basically deals with conversion of readable plain text to a complex cipher text through certain transformations involving the Input Key (Password) and some constants. In 128 bit AES encryption scheme, both the Plain text and the key are arranged in the form of a 4X4 matrix. Each element of the matrix is 1 byte in size. The transformations are applied on these matrices to obtain the Cipher text. The basic principles of encryption are Diffusion and Confusion. Diffusion principle, transforms each character of the plain text to one of many other valid characters. Confusion principle, on the other hand reorders the characters of the plain text. In AES these principles are adapted into the transformations. Each of such transformation is repeated several times sequentially in order to increase the complexity of the cipher.

AES ALGORITHM

It is also called as the Rijndael algorithm [1]. It is available in 3 variants of 128, 192 or 256 bits. It generates its round key from an input key using the Key Expansion function. The state undergoes 4 transformations namely the Shift Row, Mix Column, Add Round Key and Sub Byte transformation.

A. Shift Row Transformation

The rows of the state matrix are cyclically shifted but each row is shifted with a different offset. The offset of right shift varies from zero to three bytes. Since this transformation doesn'tcontribute to the complexity of the cipher text, we leave this transformation unchanged.

B. Mix Column Transformation

It is a linear byte substitution transformation. Here we replace the elements of the state matrix by mixing them with a constant matrix. The state matrix is multiplied with a constant matrix to obtain the new matrix. Matrix multiplication is done over Galois Field. In this transformation, the bytes are treated as a polynomials rather than numbers. The implementation complexity of this transformation is high and hence this algorithm is also used as it is.

C. Add Round Key

This Transformation involves a bitwise XOR operation between the state array and the resulting

Corresponding author: Manohar Nagallapati

Round Key that is output of the Key Expansion algorithm.

D. Sub Byte Transformation

Sub Byte transformation is a non-linear byte substitution transformation, unlike the Mix Columns where each element of the state matrix is replaced by a new element from a look up table. The main complexity of the algorithm lies in this transformation. The implementation of this transformation is very simple and hence all the modifications presented in this paper are on this transformation. The modifications increase the cipher complexity by a huge volume and marginally increase the implementation complexity.

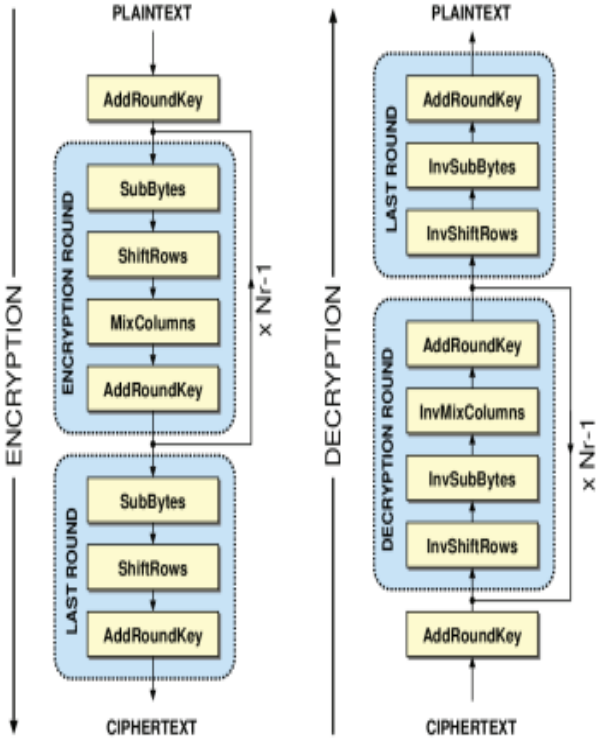


Fig 1: AES encryption & decryption flowchart.

PROPOSED DYNAMIC DUAL KEY BASED AES

In this paper we propose a slightly modified version of the algorithm in [2] and the architecture used for the efficient implementation of the algorithm.

Key-Based S-BOX

Developments on the algorithm have suggested probabilistic techniques for key based S-Box generation are highly efficient but they are extremely complex for hardware implementation. Hence the algorithm proposed in [3] for the generation of S-Byte is equally efficient in terms of complexity of reverse engineering but very simple to understand and implement. The algorithm for generating key based S-Box is shown below. It involves reordering of the S-Box by an offset which is generated from the input key.

mod_val= (key [127:120]) %8
for: i=0 to 15
Loop
Temp[i] =key [(8i) +mod_val]
For: j=0 to 7
loop
Offset[j] =temp [2i] +temp [2i+1]
dyn_subbyte = static_subbyte ^ offset

The algorithm used for AES is as shown below:

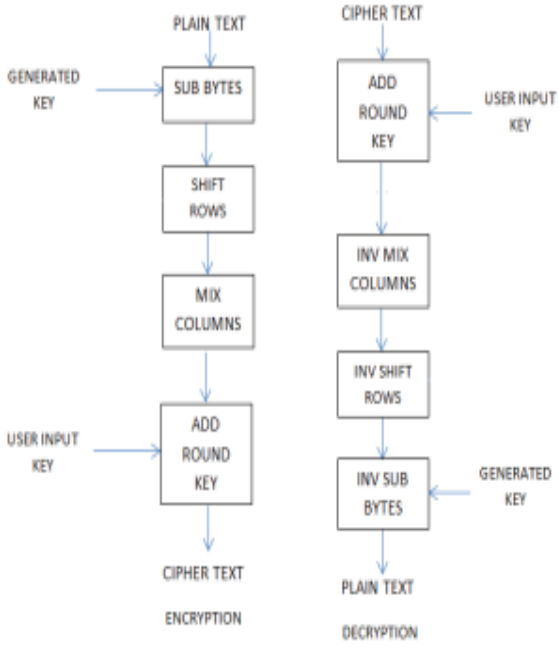


Fig 5: Block Diagram of Dual Key Based AES

The proposed algorithm involves the generation of key based S-Bytes as suggested in [3] and [4]. The two keys used for the algorithm are called User Key and System Key. System Key is generated within the system and User key is input to the system by the user. The user inputs an 8-bit value called SEED for the generation of System Key. 16 pre-defined keys are stored in the form of a look up table. A 4 bit offset is generated from the SEED to select one of the keys as System Key. But the pre-defined keys that are stored are of 120 bit length. The generation of the 128 bit System Key happens as follows:

- A 4 bit offset is generated by bitwise XOR of higher nibble and the lower nibble.
- The 120 bit key corresponding to the location of the offset value is selected.
- Append the value of the SEED to the 120 bit Key to obtain a 128 bit SYSTEM KEY

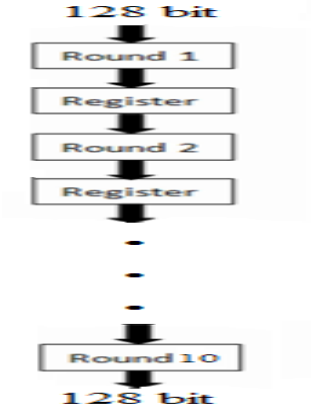
The System Key is used for the generation of S-Byte and the user key is used for the ADD ROUND KEY transformation. The S-Byte transformation used for ADD ROUND KEY is the conventional static S-Byte. This is done in order to avoid complexity during the decryption process. The System key for the next round will be obtained by bitwise XOR of the Next Round Key with the System Key.

This algorithm removes the computational overheads in [2] that do not add to the complexity of the cipher and adds simple transformations that add to the complexity of the cipher.

DUAL KEY-BASED AES WITH PIPELINING

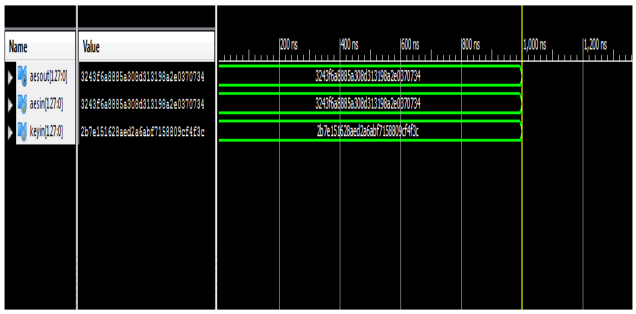
The goal of Dual key-based AES implementation using pipelined architecture is to optimize the delay. The block diagram of Dual key-based AES implementation using pipelined architecture basically two types of pipelining are possible in Dual key-based AES implementation: inner round pipelining and outer round pipelining. Here in our design we have used the outer round pipelining. Here pipelining is done after each round hence it is called outer round pipelining and also pipelining is done in non-feedback mode so it is called pipelined Dual key-based AES. In pipelined Dual key-based AES pipeline registers are placed after eachround.

The pipelining between each round will achieve high performance implementation both at encryption and decryption side. Although implementing an iterative pipelining based approach is one option, for simplicity and clarity, we have used the fully expanded implementation for all ten rounds. The data generated in each individual round is utilized as an input data to next round. Here placing the pipeline registers is the key of achieving better performance. Pipeline registers are basically used for intermediate data processing.

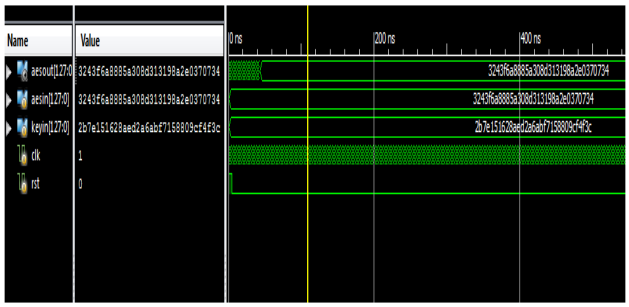


SIMULATION RESULTS

Dual key-based AES:-



Dual key-based AES with pipelining:-



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. Static S-Boxes are implemented using look up tables which will never vary with the input text or input key. This technique makes reverse engineering very simple for the purpose of cryptanalysis. Another weakness of AES is that it works with a single key. The confidentiality of the key determines the security of the algorithm .A new scheme of AES involving generation of Key based S-Boxes and dual key AES is proposed. We will be doing the outer round pipelining for all the 10 rounds in it to optimize the delay. The AES algorithm implemented using Xilinx 14.7.

REFERENCES

1. "Advanced Encryption Standard, Federal Information Processing Standards 197", National Institute of Standards and Technology, November 2001
2. Abhiram.L.S, Gowrav.L, Punith Kumar.H.L, Sriroop.B.K, Manjunath C Lakkannavar, "Design and Synthesis of Dual Key based AES Encryption", IC2014 conference, MSRIT, 2014

3. Abhiram.L.S, Gowrav.L, Punith Kumar.H.L, Sriroop.B.K, Manjunath C Lakkannavar, “Design and synthesis of Pseudo dynamic S-BOX based AES encryption”, National Conference on Electrical and Electronics Engineering, HKBK college of Engineering, 2014
4. A.F Webster and S.E Travares, "On the Design of S-boxes," Queen's university Kingston, Springer-verilog, Canada1998.
5. F. Fahmy and G. Salama, "A proposal for Key-dependant AES, " 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT), TUNISIA March 2005.
6. Joan Daemen and Vincent Rijmen.“Two-Round AES Dierential”. Cryptology ePrint Archive, Report 2006/039, 2006

Authors Details:

	**Manohar Nagallapati** has completed his B.Tech in Electronics and Communication Engineering from Global College of Engineering and Technology (affiliated to J.N.T.U.A) Kadapa in 2013. He is pursuing his M.Tech in VLSI from Vaagdevi Institute of Technology and Science (affiliated to J.N.T.U.A) Proddatur.
	**K. Naveen Kumar Raju** is an Assistant Professor at Vaagdevi Institute of Technology & Science, Proddatur. He received his Bachelor degree in Electronic and Communication Engineering from AITS (affiliated to J.N.T.U.H) Rajampet, M.Tech, Embedded Systems from AITS (affiliated to J.N.T.U.A) Rajampet. His research interest is Low power VLSI Design