Original Research Paper Engineering

FPGA IMPLEMENTATION OF DUAL KEY BASED AES ENCRYPTION AND DECRYPTION WITH KEY BASED S-BOX GENERATION

Yada Indumathi M.Tech VLSI Design, Anurag Group of Institutions (Autonomous), Hyderabad, Telangana, India.

M.Shiva Kumar Assistant Professor, Dept of VLSI, Anurag Group of Institutions (Autonoums), Hyderabad, Telangana, India.

ABSTRACT AES is mostly and approved algorithm in cryptography. Cryptography is an important role in secure network. The important secure network block cipher is Rijndael cipher and also known as AES. The advanced research is going in the field of cryptography. Finally a most of changes have been proposed on this AES algorithm. Static S-Boxes in this algorithm are implemented using look up tables in this they did not vary in input key or input text so this technique is easy to reverse engineering. This consumes a lot of space for the look up table. Another disadvantage of AES is that it works with a single key. So it is essential to generate S-Bytes at simulation time. It is beneficial if the S-byte generated during run time varies with the input key. In this paper, a new theory of AES is proposed, dual key AES with Key Based S-Box Generation. In this increases the level of security by using high usage of resources and less power with high performance. In this, the architecture of the algorithm for optimal FPGA implementation using Verilog HDL code using Vivado 14.3 and implemented on Zynq Board (FPGA).

KEYWORDS : AES, S-Box, Dual key, FPGA.

I. INTRODUCTION:

Lots and Lots of modification have been done on AES encryption. Some of them have come across to light and some are not. The ones who published previously, the complexity of the algorithm is generation of S-boxes. This also makes the hardware implementation of the algorithm very complex. The level of security provided by the tradeoff with the amount of resources consumed. An efficient architecture provides a high level of security consuming minimal resources. In this paper we present an algorithm whose implementation on an FPGA consumes minimal resources. AES consists of 128 block length of bits and supports 128, 192 and 256 key length bits. The 128 bits are organized into state matrix which is of the size of 4×4. This algorithm starts with initial transformation of state matrix Followed by nine iteration of rounds. A round consists of four transformations: Byte Substitution (subbytes), Row Shifting (shiftrows), Mixing of columns (mixcolumns) and followed by addition of Round Key called (addroundkey). From each round, a round key is generated from the original key through key scheduling Process.The last round consists of subbytes, shiftrows and addroundkey transformation.Subbytes transformation is implemented using S-Box.The S-Box is one of most time consuming process because it is required in every round.

This paper begins with a brief overview about the conventional AES encryption and also about its vulnerabilities. We move on to discuss about the recent modifications that have been done on the AES scheme and their weaknesses. Later we propose our algorithm and its FPGA implementation. We will then prove our algorithm by describing the simulation results.

II. EXISTING MODEL:

AES is a specification for the encryption of electronic data. It has been adopted by the U.S. government and is now used worldwide. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The block size has a maximum of 256 bits, but the key size has no theoretical maximum. AES operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field. The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

The algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. Cipher converts data, commonly known as plaintext, to an unintelligible form called cipher. Key Expansion generates a key schedule that is used in the Cipher and the Inverse Cipher procedure. Cipher and Inverse Cipher are composed of specific number of rounds. For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key length.

.

A. Shift Row Transformation: Shift Rows is a cyclic shift operation in each row of the State. In this operation, the bytes in the first row of the state do not change. The second, third, and fourth rows shift cyclically to the left one byte, two bytes, three bytes, respectively, as illustrated in Figure. The reverse process, inv Shift Row, operates in reverse order to Shift Rows.

B.Mix Column Transformation: The Mix Column transformation is performed independently on the state Column-by-column. The columns of the state are considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial c(x), given by:

$$c(x)=\{03\}x3 + \{01\}x2 + \{01\}x + \{02\}.$$

C.Add RoundKey Transformation: In the AddRoundKey transformation, a Round Key is added to the State - resulted from the operation of the Mix Columns transformation - by a simple bitwise XOR operation. The RoundKey of each round is derived from the main key using the Key Expansion algorithm. This transformation is the same for both encryption and decryption AddRoundKey is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order.

D. Sub byte transformation: The Sub Bytes operation is a nonlinear byte substitution. Each byte from the input state is replaced by another byte according to the substitution box (called the S-box). The S-box is computed based on by a new element from a look up table. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. 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.

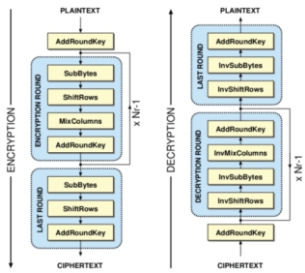

Fig 1: AES Encryption and Decryption Flowchart

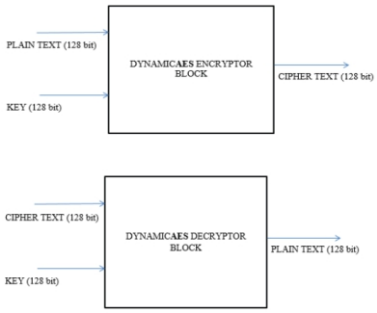

Fig 2: Block Box Representation of AES Encryption and Decryption

All these operations are repeated 10 times in an AES-128 scheme and the block diagram is shown in Fig 2 and the black box representation is shown in Fig 2.The decryption scheme retraces the steps performing the inverse of every transformation to obtain the plain text.

III. PROPOSED SYSTEM

It is mentioned that there are two ways of generation of S-Bytes.
1. Look up tables.
2. Dynamic generation of S-Bytes.

The first method consumes a lot of memory in order to store the look up table for each byte. The second method consumes significantly less area but marginally adds to the performance overhead. In this paper we propose a slightly modified version of the AES algorithm and the architecture used for the efficient implementation of the algorithm. The proposed algorithm involves the generation of key based S-Bytes. 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 algorithm used for AES is as shown below:

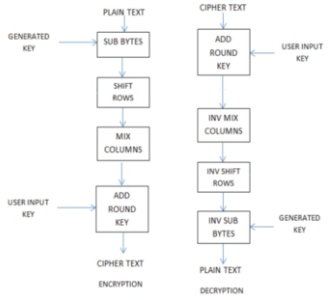

Fig 3: Block Diagram of Dual Key Based AES

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 predefined keys that are stored are of 120 bit length. The generation of the 128 bit System Key happens as follows:

1. A 4 bit offset is generated by bitwise XOR of higher nibble and the lower nibble.
2. The 120 bit key corresponding to the location of the offset value is selected.
3. 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 that do not add to the complexity of the cipher and adds simple transformations that add to the complexity of the cipher.

Thus we can easily justify that the level of security will increase by an order of 38. Also the complexity of cryptanalysis increases by an order of 4.But with dual key, the algorithm has to work backwards 4 times in order to obtain both the keys. Thus the complexity of cryptanalysis increases by 4.

VI. SIMULATION AND RESULTS:

A. Simulation Results:

Verilog is used as the hardware description language because of the flexibility to exchange among environments. The software used for this work is Vivado Tool and the waveforms are simulated with the help of Vivado. Simulator. This is used for writing, debugging, simulating and checking the performance results using the simulation tools available on Vivado.Simulation result of the AES algorithm is shown in following fig. which shows the encoding & decoding of the data which is done by using active hdl.

Encryption:

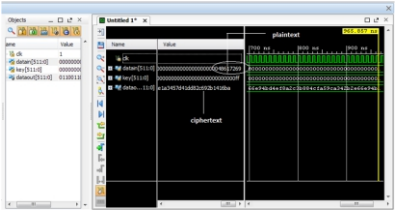

Fig 4: Simulation result of Encryption

In the above simulation inputs are plain text and user key 512 bit and the final output is cipher text 512 bit.

Decryption:

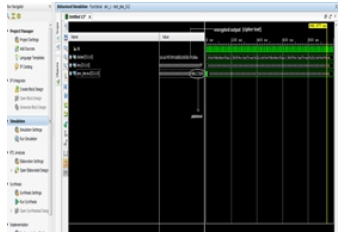

Fig 5: Simulation result of Decryption

In the above simulation inputs are cipher text and user key 512 bit and the final output is plain text 512 bit.

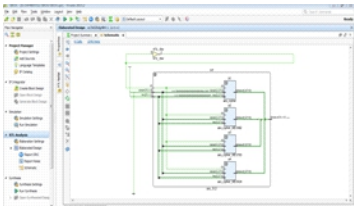

Fig 6: RTL Schematic view of Encryption

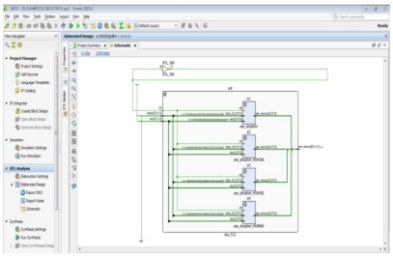

Fig 7: RTL Schematic view of Decryption

B. Hardware Implementation Results

ZYNQ Board (FPGA) 7000 series was implemented in family XC7Z020 with package number CLG484 and speed grade is -1.Vendor is Xilinx.

Encryption:

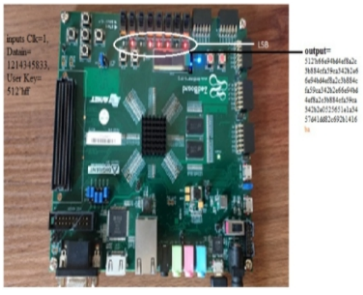

Fig 8: Result showing Encryption (LSB)

Above implementation we are considering only LSB bits of output.

Decryption

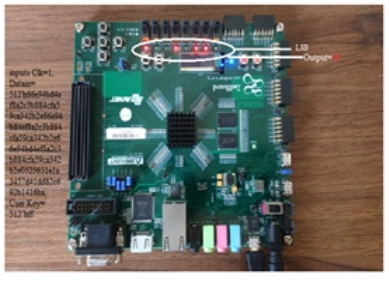

Fig 9: Result showing Decryption (LSB)

Above implementation we are considering only LSB bits of output.

V. RESULT ANALYSIS:

In our test case we have taken 512 bit data by the key
Input = [512'd 1214345833].
Cipher key= [512'dff]
Then after ten rounds of the AES the cipher text will appear as
Ciphertext=
[512'h66e94bd4ef8a2c3b884cfa59ca342b2e66e94bd4ef8a2c3b884cfa59ca342b2e66e94bd4ef8a2c3884cfa59ca342b2e0525651e1a3457d41dd82c692b1416ba]

For decryption we use cipher text as input and same cipher key for decryption algorithm and find original data. Below utilization report shows the that usage of resources are increased by that the complexity is increases due the security also increases

RESOURCE UTILIZATION

Logic Utilization	Used	Available	Utilization
Number of Slices	5632	106400	5.29
Number of slices of FF	36612	53200	68.82
Number of inputs LUT's	1537	200	768.50
BUFG	1	32	3.12

TABLE – 1: Resource Utilization

ADVANTAGES:

- High security
- Low power consumption

APPLICATIONS:

- Digital signal processing

Data transmission

VI. CONCLUSION:

The proposed dual key based AES algorithm of 512bits was simulated for a clock frequency of 1GHz and the output was obtained in 40 clock cycles. The minimum time required for the Done signal to be high from the time at which inp and start is pulled high is around 40ns. From the results we see that the marginal increase in resource utilization and a marginal decrease in the performance contribute to a significant increase in the level of security. So that marginal increase in resource utilization it shows the increasing level of security to high.

REFERENCES:

[1] 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
[2] Abhiram.L.S, Gowrav.L, Punith Kumar.H.L, Sriroop.B.K, Manjunath C Lakkannavar, "Design and synthesis of Pseudo dynamic S-BOX basedAES encryption", National Conference on Electrical and Electronics Engineering, HKBK college of Engineering, 2014
[3] A.F Webster and S.E Travares, "On The Design of S-boxes,"Queen's university Kingston, Springer-verlag, Canada1998 .
[4] Muhammad Asim, "Efficient and Simple Method forDesigning Chaotic S-boxes, "Electronic and Telecommunications Research Journal, University of Technology Petronas, Malaysia February 2008.
[5] Xinmiao Zhang and Keshab K. Parhi, "On the Optimum Constructions of Composite Field for the AES Algorithm", IEEE, VOL. 53, NO. 10, OCTOBER2006.
[6] Joan Daemen and Vincent Rijmen."Two-Round AES Dierential". Cryptology ePrint Archive, Report 2006/039, 2006