International Journal of Advanced Education and Research
ISSN: 2455-5746, Impact Factor: RJIF 5.34
www.newresearchjournal.com/education
Volume 1; Issue 5; May 2016; Page No. 08-10

Design of high secure and efficient FSR based LBIST cryptographic system

[1] Suresh K, [1] Aswani V, [1] Sandeep P, [1] Manasa Jyothi M, [2] Archana BT

[1] B.Tech, Department of Electronics and Communication Engineering, VITS College of Engineering, Visakhapatnam, Andhra Pradesh, India.

[2] Assistant Professor, Department of Electronics and Communication Engineering, VITS College of Engineering, Visakhapatnam, Andhra Pradesh, India.

Abstract
Now a day's most of the users are using wireless communication for fast sending and receiving the mails in less time and in less cost. The main issue in this way of communication is information hacking. Here a crypto device with low complexity and high security is designed by using Advanced Encryption Standard Algorithm along with Built in Self-Test technique. This paper provides the complete step by step implementation of Advanced Encryption Technique, i.e. encrypting and decrypting 128 bit data using the AES by providing enhanced reliability and security. Extra cost in terms of area is very low compared to other techniques. Because only one AES core will be originally embedded in the system. This reduces the reduction of test cost will lead to the reduction of overall production cost & 100% security of data.

Keywords: AES, LBIST, encryption

1. Introduction
Now a day's most of the users are rapidly using wireless communication technology, and in this wireless communication they are both advantages and disadvantages. This paper is proposing a method to provide more secure data by implementing Advanced Encryption Standard (AES) which specifies a FIPS-approved cryptographic algorithm that can be 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; decrypting the cipher text converts the data back into its original form, called plaintext.

Cryptographic System
Cryptography is the science of information and communication security. Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use [1]. Generally speaking, it uses a cryptographic system to transform a plaintext into a cipher text, using most of the time a key.
Caesar-cipher that obscures text by replacing each letter with the letter thirteen places down in the alphabet. Since our alphabet has 26 characters, it is enough to encrypt the cipher text again to retrieve the original message.

2. Proposed System
The paper mainly focus about security and efficiency. For that AES, LBIST algorithms are used. The AES algorithm is a symmetric block cypher that can encrypt and decrypt information. It can handle data up to 128,192,256 bits and key up to 128,192,256.In AES algorithm here 10 rounds of operations is performed. For each round different keys are generated using following operations.

Advanced Encryption Standard
The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data [2, 3]. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. This standard specifies the Rijndael algorithm, a symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits. AES is an iterated block cipher with a fixed block size of 128 and a variable key length. The different transformations operate on the intermediate results, called state. The state is a rectangular array of bytes and since the block size is 128 bits, which is 16 bytes, the rectangular array is of dimensions 4x4. After an initial Round Key addition, the State array is transformed by implementing a round function 10, 12, or 14 times (depending on the key length), with the final round differing slightly from the first Nr -1

2.1 AES Operations
2.1.1 Row Shift
In this operation, each row of the state is cyclically shifted to the left, depending on the row index.
- The 1st row is shifted 0 positions to the left.
- The 2nd row is shifted 1 position to the left.
- The 3rd row is shifted 2 positions to the left.
- The 4th row is shifted 3 positions to the left

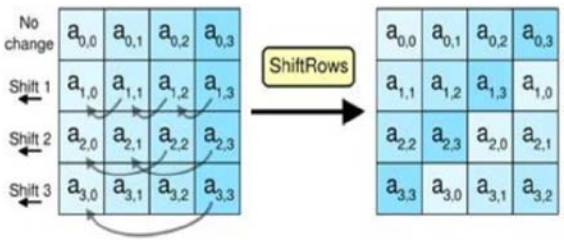

Fig 1: Row shift operation

2.2 Substitutions

The Sub Bytes operation is a non-linear byte substitution, operating on each byte of the state independently. Since the S-Box is independent of any input, pre-calculated forms are used, if enough memory (256 bytes for one S-Box) is available. Each byte of the state is then substituted by the value in the S Box whose index corresponds to the value in the state:

$$a\ (i,j) = S\ Box[a(i,j)] \qquad \text{---(1)}$$

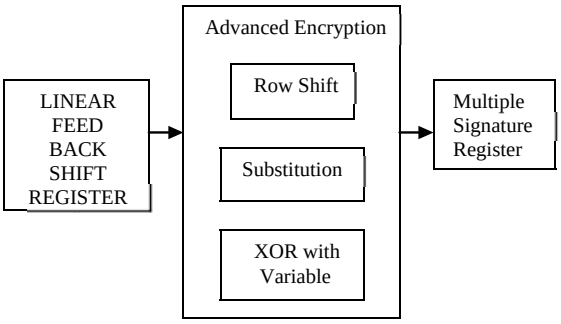

Fig 2: Block Diagram

3. XOR with Variable

In this operation, a Round Key is applied to the state by a simple bitwise XOR. The Round key is derived from the Cipher Key by the means of the key schedule. The Round Key length is equal to the block key length (=16 bytes).

It is very important to know that the cipher input bytes are mapped onto the state bytes in the order *a0,0, a1,0, a2,0, a3,0, a0,1, a1,1, a2,1, a3,1*... and the bytes of the cipher key are mapped onto the array in the order *k0,0, k1,0, k2,0, k3,0, k0,1, k1,1, k2,1, k3,1*... At the end of the cipher operation, the cipher output is extracted from the state by taking the state bytes in the same order. AES uses a variable number of rounds, which are fixed: A key of size 128 has 10 rounds. A key of size 192 has 12 rounds. A key of size 256 has 14 rounds.

During each round, the following operations are applied on the state:

1. Sub Bytes: every byte in the state is replaced by another one, using the Rijndael S-Box
2. Shift Row: every row in the 4x4 array is shifted a certain amount to the left.
3. Mix Column: a linear transformation on the columns of the state.
4. Add Round Key: each byte of the state is combined with a round key, which is a different key for each round and derived from the Rijndael key schedule.

Implementation of AES is performed with the Cipher Key expansion [4, 5]. Input cipher key is expanded, whose size varies between 128 and 256 bits into a larger key, from which different Round Keys can be derived. The S-Box values can either be calculated on-the-fly to save memory or the pre calculated values can be stored in an array. It is known already that Rotate takes a word (a 4-byte array) and rotates it 8 bit to the left. Since 8 bit correspond to one byte and our array type is character (whose size is one byte), rotating 8 bit to the left corresponds to shifting cyclically the array values one to the left

4. Result

Here AES-based cryptographic core system is implemented using the Xilinx software. The data size considered here is 128 bits three addition modes are added to the current mission of the AES crypto core. One for pseudo- random test pattern generation & one for signature analysis. Efficiency of these three modes has been demonstrated. Extra cost in terms of area is very low compared to other techniques. Because only one AES core will be originally embedded in the system. This reduces the reduction of test cost will lead to the reduction of overall production cost & 100% security of data.

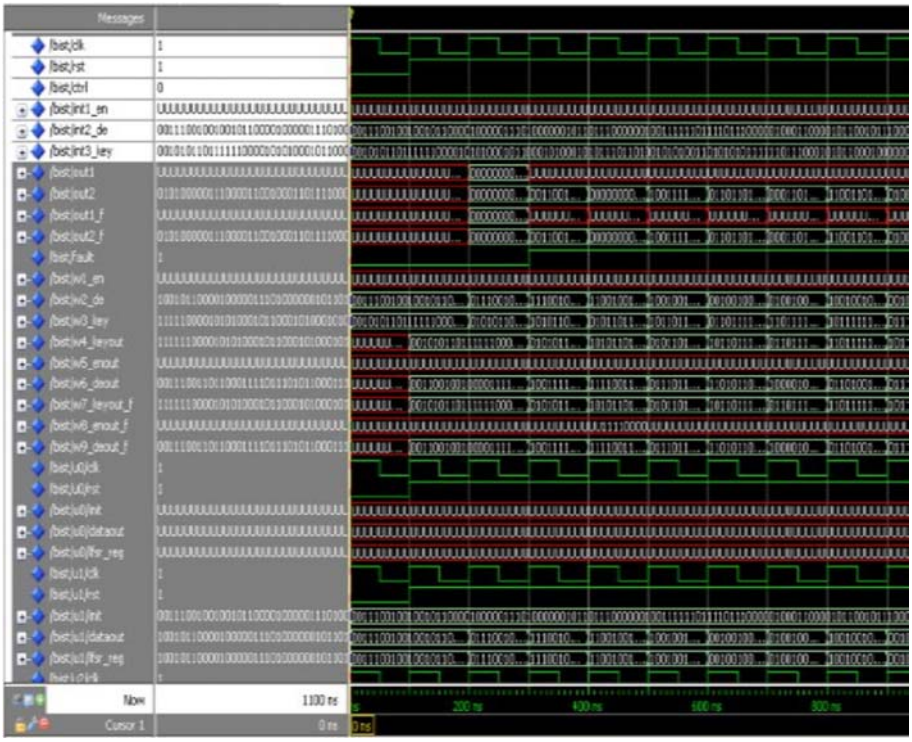

Fig 3: AES algorithm simulation output.

5. Conclusion

Various data messages were encrypted using different keys and varying key sizes. The original data was properly retrieved via decryption of the cipher text. The modifications brought about in the code was tested and proved to be accurately encrypting and decrypting the data messages with even higher security and immunity.

6. References

1. Hans Dobbertin, Vincent Rijmen, Aleksa ndra Sowa. Advanced Encryption Standard – Aes, Springer Science & Business Media, 2005.
2. The First 10 Years of Advanced Encryption. Copublished by the IEEE Computer and Reliability Societies, IEEE, 2010, 72-74.
3. Ashwini M Deshpande, Mangesh S Deshpande, Devendra N Kayatanavar. FPGA Implementation of AES Encryption and Decryption, International Conference on Control, Automation, Communication and Energy Conservation, 2009.
4. Chehal Ritika, Singh Kuldeep. Efficiency and Security of Data with Symmetric Encryption Algorithms. International Journal of Advanced Research in Computer Science and Software Engineering. ISSN: 2277-128X, 2012; 2(8).
5. Joan Daemen, Vincent Rijmen. The Design of Rijndael: AES - The Advanced Encryption Standard, Springer Science & Business Media, 2002.