International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), Volume 5, Special Issue 2, May 2015)
International Conference on Advances in Computer and Communication Engineering (ACCE-2015)

Logic Implementation of S-Box

Sandhya.N [1], Dr. Aswatha Kumar M.[2]

[1]*Student (M.tech), Sapthagiri College of Engineering, Bangalore- 57.*
[2]*Principal, Sapthagiri College of Engineering , Bangalore- 57.*

[1]sandhyanarayan15@gmail.com, [2]principal@sapthagiri.edu.in

***Abstract*— The symmetric key standard for encryption and decryption is Advanced Encryption Standard (AES). Verilog code is used to design and synthesize 128 bit AES using Rijndael algorithm.. In the proposed AES, a composite field S-Box is implemented using logic gates and divided into five blocks. The S-box is implemented using a multiple inverse logic. It consist of many blocks using which the equivalent substitution of the input is carried out in substitution block of AES algorithm.**

***Keywords*—Advanced Encryption Standard (AES), encryption, decryption, Rijndael.**

I. INTRODUCTION

The high growth in the networking technology leads a common culture for data to be re-distributed by hackers. The information has to be protected while transmitting it, sensitive information related to credit cards, banking transactions and social security numbers need to be protected. For this purpose many encryption techniques are existing which are used to avoid the theft of information. In wireless communication encryption of data plays a major role. The focus is mainly on the security of online data transmission. Different encryption techniques are used to protect the confidential data from unauthorized use. The information security can be carried through a very common technique encryption. The evolution of encryption is moving towards a future of endless possibilities

II. LITERATURE SURVEY

Security algorithms have been used from long time. The evolution of these algorithms is described as below.

T. Subashri et al [1], describes one of the best symmetric security algorithms to provide data security in AES. The pipelined architecture of the AES algorithm increases the throughput of the algorithm and the pipelined key schedule algorithm increases the speed. In this architecture, instead of passing the output of each round to the next round directly, a register is used.

It keeps off the direct contact between two rounds. With the help of search based look-up table, the hardware cost is reduced. The speed of the AES is increased.

The algorithm described by AES is a symmetric key algorithm in which the same key is used for encrypting and decrypting the data. AES is based on substitution permutation network design principle and its execution is fast in both software and hardware. It does not use a feistel system. AES is a modification of Rijndael which has a fixed block size of 128 bits and a key size of 128,192 or 256 bits. It operates an 4x4 column major order matrix of bytes termed the state although some version of Rijndael have a larger block size have additional columns in the state. AES transformations are carried in a special finite field. In the AES, the cipher text is generated after 10 rounds where encryption round consists of 4 transformations which are add round key, sub bytes, shift row and mix column transformation. The decryption algorithm transforms the cipher text to the original plain text using the reverse procedure. In AES transformation, only the S-boxes in the encryption and inverse S-boxes in decryption are nonlinear.

The receiver with a specific key can only be able to retrieve the original data when using AES in transferring data but reliability of data transfer is not confirmed because of defects in implemented structure of AES or interventions of attackers. Fault detection is an inevitable part of AES hardware implementation because of two reasons, natural faults caused by defects in gates may result in erroneous output in encryption/decryption. Attackers can also inject certain faults in AES to retrieve the key & break the system.

J. Vijaya and M. Rajaram [2], describes a fixed coefficient multiplier for mix column operation and an equivalent pipelined architecture that leads to effective utilization of resources and increase in speed. High throughput 128 bit AES is achieved by using pipelined architecture.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), Volume 5, Special Issue 2, May 2015)
International Conference on Advances in Computer and Communication Engineering (ACCE-2015)

The speed is further enhanced by inserting compact and flexible architecture for mix column transform. A suggestion is made on fixed coefficient multiplier for mix column operation and an equivalent pipelined AES architecture by changing the inner process order in round transformation which leads to effective utilization of resources and increase in speed. The proposed pipelined algorithm achieves a high data throughput when compared with other methods.

Priyanka Pimpale et al [3], a method is used in which the complexity of the encryption is decreased making it complicated for the attacker to predict a pattern in the algorithm. In each transformation of the modified architecture, the 8-bit values are separated in to 4-bits and they are grouped and then perform the transformation process. The modifications have provided the algorithm with strong diffusion and confusion.

Aluned H. Sawahneh [4], high data throughput AES hardware architecture is proposed by partitioning the 10 rounds into sub blocks of repeated AES modules. To provide a complete ten stages of AES, the intermediate buffers are used to separate the blocks. Using this pipelined architecture scheme, time complexity is reduced to higher extent.

K. Rahimwmisa et al [5], a simple, linear and cryptanalysis is carried on the standard S-box to take advantage of high probability occurrences of linear expressions involving plain text bits, cipher text bits and sub-key bits. The operation of the cipher is linear where the linearity refers to a mod-2-bitwise operation. The design can run at 1.2 GHz which is sufficient for online data encryption. To increase the performance of the executed circuit, particularly cost and power, all of the AES blocks may be reconfigurable. So the parameters used for reconfiguration are implanted inside the manager module of reconfiguring and also possible to quickly cross from a safe configuration to another by updating a hard system protection.

Joan Daemen et al [8], describes implementation of Rijndael to run fast for a block cipher on a Pentium. There is a trade-off between table size/performance. Rijndael can be implemented on a smart card in a small amount of code, using a small amount of RAM and taking a small number of cycles. There is some ROM/performance trade-off. It does not make use of another cryptographic component, S-boxes lent from well-reputed ciphers, bits obtained from rand tables, digits of p.

Based on variable block length the block lengths of 192 and 256 bits allow the construction of a collision-resistant iterated hash function using Rijndael as the compression function. The block length of 128 bits is not considered sufficient for this purpose. Based on the extensions of AES the design allows the specification of variants with the block length and key length both ranging from 128 to 256 bits in steps of 32 bits. Although the number of rounds of Rijndael is fixed in the specification, it can be modified as a parameter in case of security problems. The key schedule supports any key length that is a multiple of 4 bytes. The only parameter that needs to be defined for other key lengths than 128, 192 or 256 is the number of rounds in the cipher. The cipher structure lends itself for any block length that is a multiple of 4 bytes, with a minimum of 16 bytes. The key addition and the byte substitution and mixcolumn transformations are independent from the block length. The larger block length causes the range of possible patterns that can be applied at the input or output of the sequence of the rounds to increase. This added flexibility may allow to extend attacks by one or two rounds.

III. METHODOLOGY

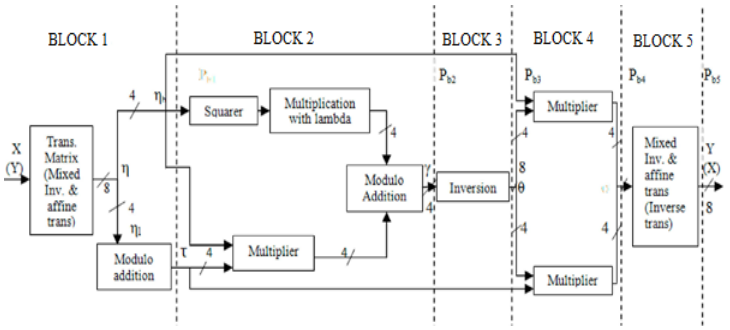

Fig 1: S-box module [9]

Figure 1 shows S-box module. The subbyte transformation is computed by taking the multiplicative inverse in GF (2^8) followed by an affine transformation. For its reverse, the invsubbyte transformation, the inverse affine transformation is applied first prior to computing the multiplicative inverse. The multiplicative inverse computation will be done by decomposing the more complex GF (2^8) to lower order fields of GF (2^1), GF (2^4) and GF (2^{2^2}).

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), Volume 5, Special Issue 2, May 2015)
International Conference on Advances in Computer and Communication Engineering (ACCE-2015)

Computation of the multiplicative inverse in composite fields cannot be directly applied to an element which is based on GF (2^8). That element has to be mapped to its composite field representation via an isomorphic function, δ. After performing the multiplicative inversion, the result will also have to be mapped back from its composite field representation to its equivalent in GF (2^8) via the inverse isomorphic function.

IV. RESULTS

The simulated result for S-box is shown in figure 2.

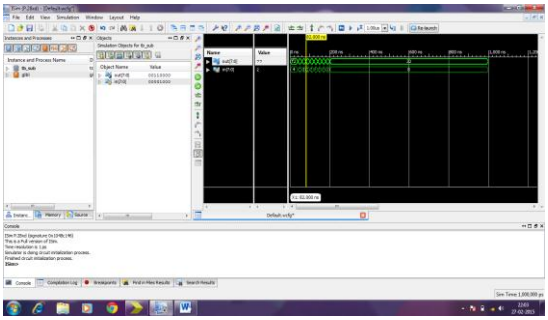

Fig 2: S-box output

V. CONCLUSION

The S-box output is calculated using multiplication inverse technique. The use of look-up table is eliminated which in turn reduces the memory usage. The equivalent value for the input is substituted by calculating dynamically, in less time. The use of this technique is very helpful as it saves time of operation. The substitution of the equivalent value increases the security of the in-out.

REFERENCES

[1] T Subashri, R Arunachalam, Gokul Vinoth Kumar B and V Vaidehi, Pipelining architecture of AES encryption and key generation with search based memory, International journal of VLSI design & Communication Systems (VLSICS), Vol 1 No.4, Dec 2010, pp 48-60.

[2] J. Vijaya and M. Rajaram, High speed pipelined AES with mix column transform, European Journal of Scientific Research, ISSN 1450-216X Vol 61 No.2, 2011, pp 47-50.

[3] Priyanka Pimpale, Rohan Rayarikar, Sanket Upadhyay, Modifications to AES algorithm for complex encryption, IJCSNS International Journal of Computer Science and Network Security, Vol 11 No.1 0, Oct 2011, pp 201-207.

[4] Aluned H. Sawahneh, Hardware design of AES S-box using pipelining structure over GF (24i), 2011, pp 346-350.

[5] K. Rahimwmisa, Dr. S. Suresh Kumar, and Rajesh Kumar, Implementation of AES with new S-box and performance analysis with the modified S-box, International Conference on VLSI, Communication & Instrumentation (ICVCI) 20J J Proceedings published by International Journal of Computer Applications® (IJCA), 2011, pp 205-210.

[6] Moo Seop Kim, Juhan Kim, and Yongje Choi, Low power circuit architecture of AES crypto module for wireless sensor network, World Academy of Science, Engineering and Technology 8, 2007, pp 1217-1221.

[7] M. Pitchaiah, Philemon Daniel, and Praveen, Implementation of advanced encryption standard algorithm, International Journal of Scientific & Engineering Research, Vol 3, Issue 3, Mar 2012, pp 1-6.

[8] G. Alisha Evangeline, S. Krithiga, S. Sheeba Ran Gnanamalar, Least Complex S-box and Its Fault Detection for Robust Advanced Encryption Standard Algorithm, 2013, pp 51-56.

[9] National Institute of Standards and Technology, Specification for the Advanced Encryption Standard (AES). FIPS PUB 197, available at http://csrc.nist.gov, 2001.