International Journal of MC Square Scientific Research Vol.8, No.1 Nov 2016

Design and Implementation of Efficient Advanced Encryption Standard Composite S-Box with CM-Mode

M.Sharmila
Applied Electronics,
Sri Balaji Chockalingam Engineering College, Arni
Dr.K.Gunasekaran
Professor/ECE
Sri Balaji Chockalingam Engineering College,Arni.
guna.k77@gmail.com

Abstract- Advanced Encryption Standard is the most popular cryptographic security algorithm used for data protection and transmission. The paper proposes an implementation of the AES new Mix- Column operation. In this paper, an enhanced Mix- Column is designed for AES decryption through Very Large Scale Integration (VLSI) System design environment. In AES Mix-Column, large number of logic gates used to perform the multiplication of input stage bytes (output of shift row) and fixed defined state bytes. In order to decrease this problem, the redundant function of Mix-Column is eliminated and re-designed in this paper. Proposed model of Mix-Column minimizes 25% of logic gates compared with previous work. Further, the proposed Mix-Column of AES decryption achieves by improving the performance of area, delay and power consumption. The implementation of the transformation is optimized and increase speed.

Keywords: Advanced Encryption Standard (AES), Rijindael Algorithm, Enhanced Mix-Column, S-box, Composite Field Arithmetic (CFA), Very Large Scale Integration (VLSI).

1. Introduction

The Advanced Encryption Standard is one of the most important cryptography algorithm known to date. It is designed for encryption of electronic data established by US National Institute of Standards and Technology in 2001. It is established on Rijndael Cipher which was succeeded by two Belgian cryptographers, Vincent Rijmaen and Joan Daemen. These two peoples submitted a proposal to NIST at the time of AES selection process. Rijndael is a family of ciphers [1] with different block sizes and key.

AES is having 128 bits of block size and three different cipher key lengths such as 128 bits, 192 bits and 256 bits. Finally AES is accepted by US government and it is used worldwide now.

Before AES design, Data Encryption Standard is used and it has asymmetric key. So it needs two different keys for data encryption and decryption process. But AES is symmetric key algorithm; cipher key is common for both encryption and decryption. The paper describes the AES-128 Rijndael algorithm for symmetric key encryption and its selected by the sensor networks. The performance of encryption and decryption on the 8-bit

Microcontroller [11], [12]. Then, it analyzes the efficiency of communication through the total delay per hop in sensor networks.

2. RIJINDAEL'S Algorithm

AES is a symmetric block cipher technique. It uses same cipher key for both encrypting and decrypting the message and the pain text. Based on round transformation, Key size is used in AES. Three numbers of rounds are used in AES which are 10 rounds, 12 rounds and 14 rounds for 128, 192 and 256 bits respectively. Each of a round consists of few steps. At first, the key expansion is needed for deriving the round key from cipher key. The cipher Rijndael [4] consists of an initial Round Key addition, Nr-1 Rounds, a final round. It shows the pseudo C code of Rijndael algorithm [10].

```
Rijndael (State, ExpandedKey)
 {
      KeyExpansion(CipherKey, ExpandedKey); Add Round Key (State,
      ExpandedKey);
For(i=1;i<Nr;i++)Round(State,ExpandedKey+Nb*i); FinalRound(State,
ExpandedKey+Nb*Nr);
}
```

The expansion of key can be done on beforehand and Rijndael can be specified in terms of the Expanded Key. The Expanded Key is derived from the Cipher Key and never be specified directly. There is however no restrictions on the selection of the Cipher Key itself.

```
Rijndael(State, ExpandedKey)
{
      AddRoundKey(State,ExpandedKey);
For(i=1;i<Nr;i++)Round(State,ExpandedKey+Nb*i);
FinalRound(State,ExpandedKey+Nb*Nr);
}
```

Further, S-box, ShiftRows, MixColumns and AddRoundKey methods are done in encryption process and reverse process is followed by decryption process. In those different levels of modes are available for encrypt and decrypt data with high level of performance.

3. Counter Mode

AES Counter Mode , the operation does not directly use the AES cipher block to encrypt the data like Electronic Code Block mode or Cipher Block

Chaining mode do; In this mode, it encrypts an arbitrary value called the counter' and then it XORs the result with the plain data to produce the ciphered text. The counter value is incremented by one for every successive block processed.

In this process, every message is divided into 128-bit vectors; each of these vectors is XORed with the result of encrypting the counter value correspondent to that block using an AES cipher block.

4. Implementation Of Enhanced SubByte Transformation

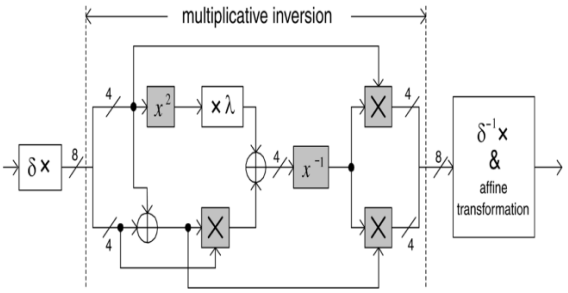

Fig. 1 Implementation of the SubBytes Transformation

The main contributions of the subbyte transformation can be reviewed as follows. Figure.1 shows implementation of SubBytes. This paper many avoid the use of LUTs and propose the use of composite field data path for the both SubBytes and InvSubBytes transformation.

Composite field arithmetic has been selected to implement efficient data paths. Instead, the proposed architecture in this paper is minimized the multiplicative inverse (MI) architecture. Further, this structure is incorporated into multiplicative inverse block of composite SubBytes transformation.

5. Composite Field Arithmetic

Composite field are frequently used in implementation of Galois field arithmetic. The non- LUT based AES algorithm is able to exploit the advantage of sub pipelining further. Nevertheless, these approaches may have high complexities of hardware. Although two Galois Fields of the same order are isomorphic, the complexity of the field operation is mainly depending on the representations of the field elements. The cost of the transformation depends on the choice of the composite field. Composite field arithmetic is used to reduce the hardware complexity. We have to implement this composite S-Box into the AES-CM getting the area and delay.

6. Existing Mix-Column Transformation For AES

The Mix-Column transformation operates on the State column-by-column and manages each column as a four term polynomial. These columns are appraised as polynomials over GF (2^8) then multiplied by modulo x^4+1 with a particular fixed polynomial x (n) is given by,

$$x(n) = \{03\}\, x^3 + \{01\}\, x^2 + \{01\}\, x + \{02\}$$

This can be written as a matrix multiplication s'(x) = x(n)*s(x), where s(x) represents the state byte from ShiftRow method. These can be presented as,

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < Nb.$$

Similarly, inverse Mix-Column can be calculated by using $x^{-1}(n)$. The columns are appraised as polynomials over GF (2^8) then multiplied modulo x4+ 1 with a particular fixed polynomial $x^{-1}(n)$.

$$x^{-1}(n) = \{0b\}\, x^3 + \{0d\}\, x^2 + \{09\}\, x + \{0e\}$$

Therefore s'(x) = $x^{-1}(n) \oplus s(x)$. The matrix for InvMix-Column can be represented as,

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < Nb.$$

The multiplication of input state bytes with fixed pre-defined polynomials can be processed by Xtime (Both input and output word length of multiplication is same) multiplication. In each multiplication steps, set of inputs are logically surrounded by EX-OR gate.

The process of Xtime multiplication for the Inverse Mix-Column transformation is illustrated in fig.2. Matrix for InvMix-Column requires more number of logic gates to

perform multiplication than Mix-Column due to long word length. Typically 24 EX-OR gates are used to perform the Inv Mix- Column operation in AES.

Due to utilizing more number of logic gates, area for existing InvMix-Column consumes large area and delay. In order to reduce this problem, circuits for InvMix-Column is realized in this paper. In next section Enhanced InvMix-Column is described in detailed manner.

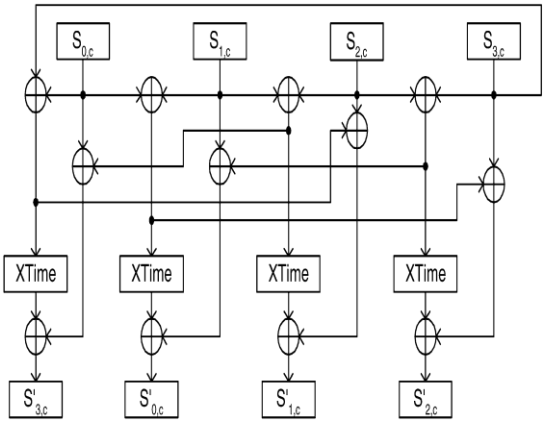

Fig. 2 Xtime Multiplication circuit for InvMix-Column Transformation

7. Enhanced Mix-Column Transformation For AES

When compared to Mix-Column transformation, InvMix-Colum transformation has multiplication of long word length.

Therefore, Matrix multiplication of InvMix- Column can be realized and re-designed. Design of Optimized InvMix-Column is represented in below. In InvMix-Column, {09, 0b, 0d, 0e} is multiplied with input bytes. Let input bytes are represented as b0 to b7 and output of InvMix-Column are represented as t0 to t7. The multiplication can be described as follows: Multiplication of {09} with state-byte,

$$t_7 = 0,\ t_6 = b_7, t_5 = b_6 \oplus b_7, t_4 = b_5 \oplus b_6,$$

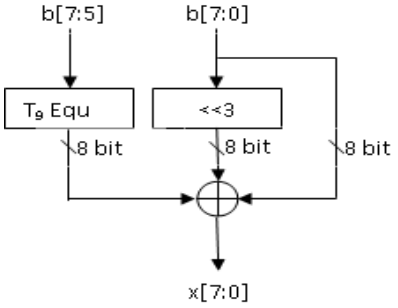

Fig. 3 Advanced Xtime for 09

Multiplication of {0b} with state-byte,

$$t_7 = 0,\ t_6 = b_7,\ t_5 = b_6 \oplus b_7, t_4 = b_5 \oplus t_5,$$

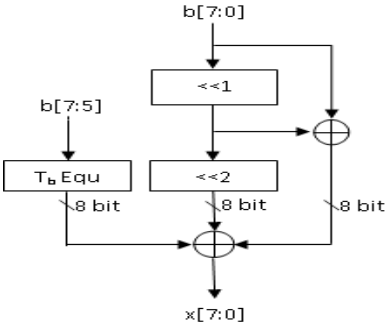

Fig. 4 Advanced Xtime for 0b

Multiplication of {0d} with state-byte,

$$t_7 = 0,\ t_6 = b_7,\ t_5 = b_6, t_4 = b_5 \oplus b_7,$$

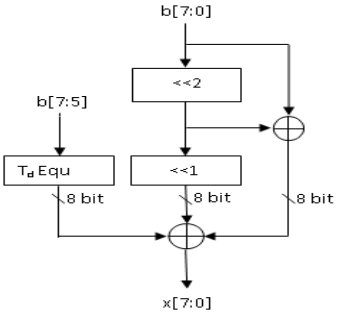

Fig. 5 Advanced Xtime for 0d

Multiplication of {0e} with state-byte,

$$t_7 = 0,\ t_6 = b_7,\ t_5 = b_6, t_4 = b_5, t_3 = b_5 \oplus b_6,$$

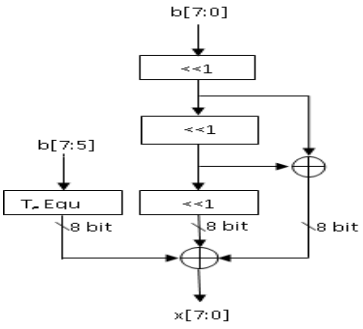

Fig. 6 Advanced Xtime for 0e

The circuit diagram for optimized Mix- Column for above equation representation is illustrated in fig. 3, fig. 4, fig. 5and fig. 6 respectively, in which only less number of EX-OR gates are used to perform the matrix multiplication.

In this optimized InvMix-Column design, 13 number of EX-OR gates are used

instead of 24 number of EX-OR gates. Hence, hardware complexity InvMix- Column can be reduced significantly than existing one. Further optimized InvMix-Column is incorporated into AES decryption process for improve the performance.

8. Implementation Of Hardware

This paper implies the detailed architecture for each nontrivial transformation in the AES Rijndael algorithm [2],[5],[8]. The design of each transformation is enhanced to reduce area and increase speed. Then efficiency for expansion of key architecture suitable for round units is proposed. Based on the analysis, the gate counts in the critical path are reduced for both the round units and the key expansion of the AES Rijndael algorithm are presented.

9. Implementation Results

The design of Optimized InvMix-Column is designed by using Verilog HDL.

The simulation results are validated using ModelSim 6.3C and synthesis results are evaluated by using Xilinx 10.1i design tool. The simulation results for AES encryption and decryption by using Optimized Mix-Column design is illustrated in fig. 7 and fig. 8 respectively.

The synthesis results for existing AES InvMix-Column and proposed AES Optimized InvMix-Column based AES decryption is analyzed and compared in Table 1.

Table.1 Comparison of Existing and Proposed AES Decryption

Types	Area	Delay	Power
Existing AES using xtime multiplication based InvMix-Column	10750	8.686	7.785
Proposed AES using Enhanced InvMix-Column	10471	3.793	6.542

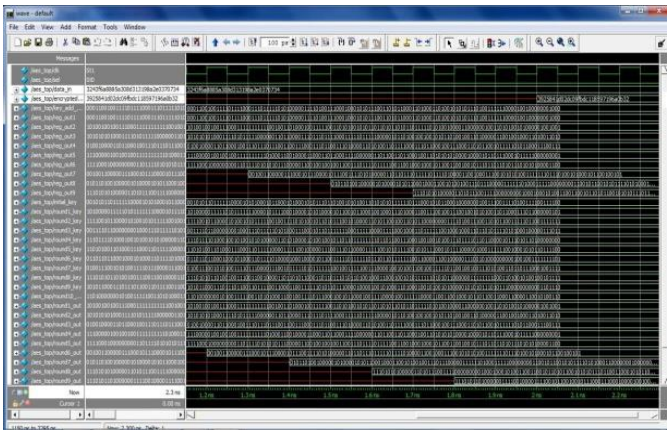

Fig. 7 AES Encryption Result

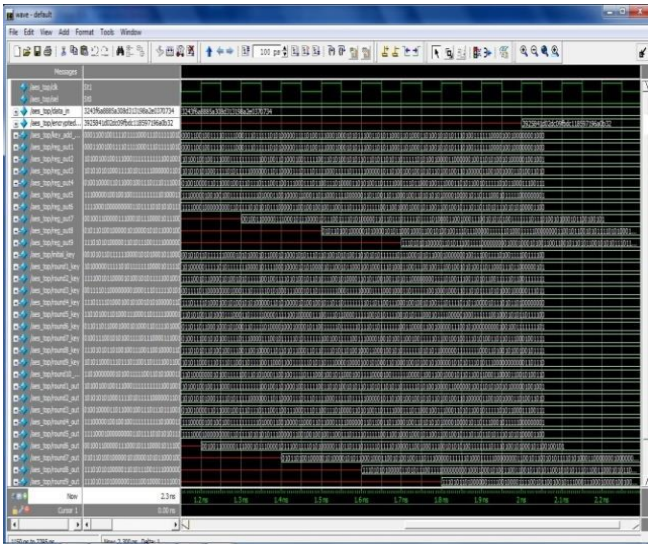

Fig. 8 AES Decryption Result

10. Conclusion

The Advanced Encryption Standard Algorithm is an iterative private key symmetric block cipher. In this paper, Enhanced InvMix-Column of AES decryption is designed through Very Large Scale Integration (VLSI) System design environment.The structure of Xtime multiplication can be optimized by reducing the logic gates. In Enhanced InvMix- Column design 9 gates are reduced to perform the Xtime multiplication. Further Enhanced InvMix- Column Transformation technique is incorporated into AES decryption for improve the performance of decryption. Proposed AES decryption gives 12.69% reduction in area, 22.90% reduction in delay and 11.90% reduction in power consumption than existing AES decryption. In future, Proposed AES decryption architecture

will be helpful in Satellite, Space, terrestrial and data communication application in order to provide security and improve the performance of hardware utilization.

REFERENCE

1. DAEMEN, J.—RIJMEN, V.: AES Proposal: Rijndael, The Rijndael Block Cipher, AES Proposal, pp.1–45, 1999 (http://csrc.nist.gov/CryptoToolkit/aes/).
2. Balamurugan, J., and Logashanmugam, E. "High speed low cost implementation of advanced encryption standard on FPGA" Current Trends in Engineering and Technology (ICCTET), 2014 2nd International Conference on. IEEE, 2014.
3. Astarloa, A., Zuloaga, A., Lazaro, J., Jimenez, J. and Cuadrado, C. "Scalable 128-bit AES-CM crypto-core reconfigurable implementation for secure communications" IEEE Applied Electronics, pp. 37-42, 2009.
4. Lee, H., Lee, K. and Shin, Y. "Implementation and Performance Analysis of AES-128 CBC algorithm in WSNs" Advanced Communication Technology (ICACT), IEEE 2010 The 12th International Conference on Vol. 1, pp. 243-248, 2010.
5. Zhang, X. and Parhi, K.K. "High-speed VLSI architectures for the AES algorithm. IEEE Transactions on Very Large Scale Integration (VLSI) Systems" Vol. 12, No. 9, pp.957-967, 2004.
6. Arbaugh, William A. "Real 802.11 security: Wi-Fi protected access and 802.11 i" Addison-Wesley Longman Publishing Co., Inc., 2003.
7. An FPGA Implementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists, in Proceedings of the Third Advanced Encryption Standard (AES) Candidate Conference, pp. 13–27 , April, 2000.
8. Hardware Evaluation of the AES Finalists, in Proceedings of the Third Advanced Encryption Standard (AES) Candidate Conference, pp. 297–285, April, 2000.
9. Verbauwhede, I., Schaumont, P. and Kuo, H. Design and performance testing of a 2.29-GB/s Rijndael processor. IEEE Journal of Solid-State Circuits, 38(3), pp.569-572, 2003.
10. Chodowiec, Pawel, Kris Gaj, Peter Bellows, and Brian Schott. "Experimental testing of the gigabit IPSec-Compliant implementations of Rijndael and triple DES using SLAAC-1V FPGA accelerator board." International Conference on Information Security, pp. 220-234. Springer Berlin Heidelberg, 2001.
11. Fu, Y., Lin H., Xuejie, Z. and Rujin Y. "Design of an extremely high performance counter mode AES reconfigurable processor." IEEE Second International Conference on Embedded Software and Systems (ICESS'05), pp. 7-pp, 2005.
12. Vu, K. and Zier, D. FPGA implementation AES for CCM mode encryption using Xilinx Spartan-II. ECE-679, Oregon State University, Spring, 2003.