Journal of Computer Applications (JCA)
ISSN: 0974-1925, Volume IV, Issue 1, 2011

Advanced Fault Detection Scheme for AES Architecture

[1]S.Anandi Reddy

Abstract— **Cryptography is a method that has been developed to ensure the secrecy of messages and transfer data securely. The Advanced Encryption Standard (AES) is the newly accepted symmetric cryptography standard for transferring block of data securely. However, the natural and malicious injected faults reduce its reliability and may cause confidential information leakage. The objective of this paper is to find optimized fault detection schemes for reaching reasonable fault coverage in the high performance AES implementations. In order to provide low cost complexity signature, two sets of error indication flag is used. Thisstructure can be applied to both look-up tables and logic gate for the implementation of S-box and inverse S-box and their parity predictions. Defects in the logic gates causedeither by the natural faults or malicious injected faults that are detected independent of the method the S-box is implemented Moreover, the overhead costs, including space complexity and time delay of the proposed schemes are analyzed. Finally, our simulation results show the error coverage of greater than 99 percent for the proposed schemes.**

Index Terms— **Advanced Encryption Standard, S-Box, inverse S-box, composite field, fault detection.**

I. INTRODUCTION

1.1 Symmetric Key Cryptography

Cryptography is a method that has been developed to ensure the secrecy of messages and transfer data securely. In digital communications the data is sent through the wires or air and thus it is not protected from eavesdropping. Therefore, confidentiality of the transferring data is of extreme importance. Encryption is a process which transforms the data that is aimed to be sent to an encrypted data using a key. The encryption process is not confidential but the key is only known to the sender and receiver of data. The receiver transforms the received data using the decryption process to obtain the original data.

Symmetric key cryptography is a form of cryptosystem in which encryption and decryption are performed using the same key. It has been utilized for secure communications for long period of time.Symmetric key cryptography comprises two different methods for encryption and decryption. It can either use stream cipher or block cipher method of encryption/decryption.

Manuscript received Jun 12, 2011.

S.Anandi Reddy, Dhanalakshmi Srinivasan Engineering College, Perambalur, India. (e-mail : anandinearu@gmail.com)

1.2 Evolution of Advanced Encryption Standard

The National Institute of Standards and Technology initiated a process to select a symmetric key encryption/ decryption algorithm in 1997. Finally, Rijndael algorithm was accepted among other finalists as the Advanced Encryption Standard (AES) in 2001.

It is noted that before the acceptance of Rijndael algorithm, DES and its improved variant 3DES were used as symmetric key standards. DES has 16 rounds and encrypts and decrypts data using a 64-bit key. 3DES has hardware implementation that doesn't produce efficient software code and three times as many rounds as DES so correspondingly slower. Both DES and 3DES use a 64-bit block size. To satisfy both efficiency and security, a larger block size is desirable.

AES-128 has 10 rounds where data is encrypted and decrypted in 128-bit blocks using a 128-bit key. It is a very good performer in both hardware and software across a wide range of computing environments.

1.3 Motivation

The objective in using AES is to transfer the data so that only the desired receiver with a specific key would be able to retrieve the original data. However, the natural and malicious injected faults reduce its reliability and may cause confidential information leakage. Thiscan be either due to:

- Natural faults caused by defects in gates or,
- Malicious injected faults to retrieve the key and break the system.

As a result, finding a suitable fault detection scheme has always been an issue in the AES. FPGAs are most flexible implementation to produces high performance withlow cost. FPGA provides more physical security with parallelism.

1.4 Objectives

The objective of this paper is to find concurrent structure independent fault detection schemes for reaching reasonable fault coverage. It makes a robust implementation of AES against these above attacks and provides highest efficiencies, showing reasonable area and time complexity overheads.

II. ADVANCED ENCRYPTION STANDARD

2.1 AES algorithm

AES is an iterated block cipher with a fixed block size of 128 and a variable key length. It hasvariable number of rounds, which is fixed according to key length. AES performs four transformations in each round in order to provide high level of security.

2.2 TRANSFORMATIONS

AES performs four transformations in each round in order to provide high level of security. This involves the properties of confusion and diffusion to provide frustrating statistical cryptanalysis. The transformations in each round of encryption except for the last round are as follows:

I. **SubBytes:** It isa non-linear substitution step where each byte is replaced with another according to a lookup table. The look table is known as S-Box which is generated by applying affine transform to multiplicative inverse of input.

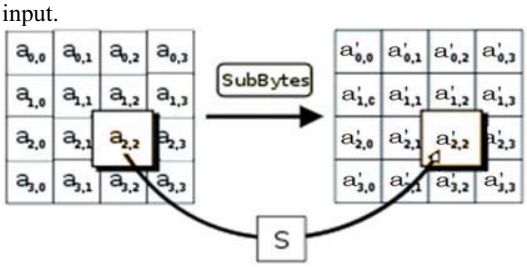

Figure.1

II. **ShiftRow:** It is a transposition step where each row of the state is shifted cyclically a certain number of steps to the left.For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. Similarly for decryption rows are shifted right.

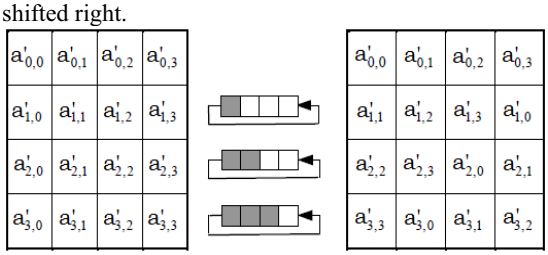

Figure.1.1

III. **MixColumn:** It is a mixing operation which operates on the columns of the state, combining the four bytes in each column using an invertible linear transformation

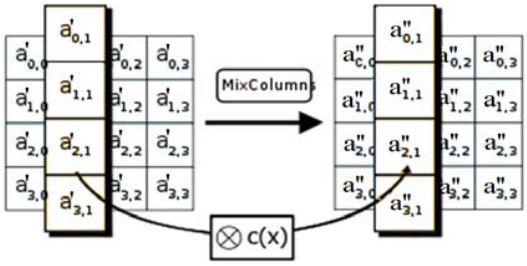

Figure.1.2

During this operation, each column is multiplied by the known matrix that for the 128 bit key is

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}.$$

IV. **AddRoundKey:** In this, each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule. The roundKey is added to the state before starting theloop. In the AddRoundKey step, each byte of the state is combined with a byte of the round sub key using the XOR operation.

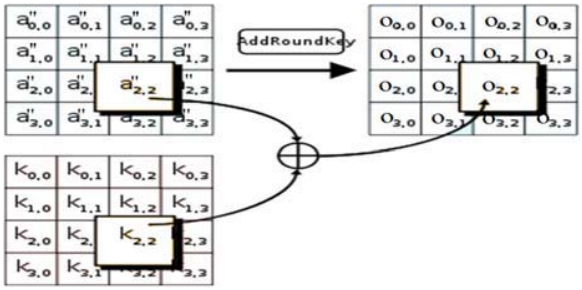

Figure.1.3

III. EXISTING SYSTEM

There has been many fault detection schemes proposed till this date to avoid the possibility of suffering from various attacks such as natural faults caused by defects in gates ,injection of fault by attackers to retrieve the key. Some of the majorly contributed schemes follow as:

3.1 A 16-BIT KEY PARITY METHOD

In this scheme, the output parity bits of each transformation in every round are predicted from the inputs. Then, the comparisons between the predicted and the actual parities (obtained using the actual parity block or predetermined parity block) can be scheduled so that the desired error coverage is obtained. Since the 128 bit input is represented in 4X4 matrix 16 parity bits corresponding to each 1 byte are compared which is presented in [3] and [12]. It has drawback that requires two blocks of 256 x 9 memory cells (S-boxes and parity predictions box). So it has relatively high area complexity for the parity predictions of MixColumns in the AES encryption. This is even more for Inv MixColumns in the AES decryption.

3.2 REDUNDANCY-BASED TECHNIQUE

The redundancy-based solution for implementing fault detection in the encryption module is based on the idea of performing a test decryption immediately after the encryption and then checking whether the original data block is obtained. The redundant unit fault detection scheme [4], [6] is used where algorithm-level, round-level, or operation-level fault detections are considered. The schemes pays time penalty either to decrypt a data block or for the comparison.

3.3 DOUBLE TIME TRANSFORMATION TECHNIQUE

In [5], the scheme uses same transformations twice in an AES round for the same data to detect the transient errors. It is time consuming and hence increases delay overhead. However, this method suffers from permanent internal faults or the malicious injected faults lasting for a long period.

3.4 MULTIPLICATION-BASED SCHEME

In [7] scheme, the result of the multiplication of the input and the output of the multiplicative inversion is compared with the predicted result of unity.

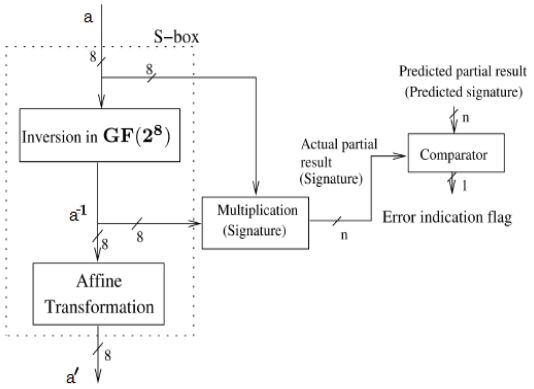

Figure. 2 The multiplication-based scheme for the fault detection of the multiplicative inversion

Since S-box is generated by applying affine transform to multiplicative inverse of an input, there is no access to the output of the multiplicative inversion. Therefore, this scheme is not suitable for the S-boxes and inverse S-boxes implemented using lookup tables (LUTs).

3.5 Pipelined Structure

Under this, pipelined distributed memories for the LUT-based S-boxes and inverse S-boxes are used to increase the design speed and the overall frequency of AES. There are three architecture to speed up, namely pipelining, subpipelining, and loop unrolling. Among these approaches, the subpipelined architecture can achieve maximum speed up and optimum speed–area ratio in non-feedback modes. Subpipelining inserts rows of registers among combinational logic not only between but also inside each round unit which is presented in [8].Non-LUT-based approaches can be used to avoid the unbreakable delay of LUTs which involve inversions in Galois Field, which may have high hardware complexities.

IV. PROPOSED SYSTEM

The proposed system detects fault while implementing a cipher from a plaintext for transmission and thus called concurrent error detection (CED).

4.1AES Encryption

The new fault detection structure for the AES encryption consist of two sets of error indication flags corresponding tocombined SubBytes and ShiftRows and combined MixColumns and AddRoundKey. A typical AES encryption round (except for the last round) consists of four transformations, and the fault detection schemes are follows.

4.1.1 SubBytes and ShiftRows

In the AES encryption, the SubBytes transformation consists of 16 S-boxes corresponding to 16 one byte of 128 bit input. Let $e_{r,c}$, $0 \le r,c \le 3$, be the error indication flag for the S-box with the input $a_{r,c}$ and the output $a'_{r,c}$.The output state of such flags can be written as 16 formulations as follows:

$$e_{r,c} = P_{(M_{r,c}a'_{r,c}+m_{r,c})} + u'_{r,c}; \quad 0 \le r,c \le 3,$$

(5)

where $u'_{r,c} = (u', 0,0,0,0,0,0,0)^T$, u is obtained by logical OR operations of all inputs and outputs of S-Box, i.e.,

$$u' = (a_0 \cup a_1 \cup \ldots . a_7) \cup (\overline{a'_0} \cup \overline{a'_1} \cup a'_2 \cup a'_3 \cup a'_4 \cup \overline{a'_5} \cup \overline{a'_6} \cup a'_7)$$

. For input hex (a) =00 and output hex (a') =63 of S-Box, u'=0 but for remaining input u'=1.And we have

$$Ma' + m = u' \quad (6)$$

where M is 8 X 8 matrix

$$M = \begin{bmatrix} a_{6,5,2} & a_{5,4,1} & a_{7,5,3,0} & a_{6,4,2} & a_{7,5,3,1} & a_{7,6,5,2,0} & a_{7,6,5,4,1} & a_{7,6,3,0} \\ a_{7,5,3,2,0} & a_{6,4,3,1} & a_{7,6,5,4,3,1,0} & a_{7,6,5,4,3,2,0} & a_{7,6,5,4,3,2,1} & a_{5,3,2,1} & a_{4,2,1,0} & a_{6,4,3,1} \\ a_{6,4,3,1} & a_{7,5,3,2,0} & a_{7,6,5,4,2} & a_{7,6,5,4,3,1} & a_{7,6,5,4,3,2,0} & a_{6,4,3,2} & a_{5,3,2,1} & a_{7,5,3,2,0} \\ a_{7,6,4,0} & a_{6,5,3} & a_{6,0} & a_{7,5} & a_{6,4} & a_{6,4,3,2,0} & a_{7,5,3,2,1} & a_{7,5,1} \\ a_{7,6,2,1} & a_{7,6,5,1,0} & a_{5,3,1} & a_{4,2,0} & a_{3,1} & a_{6,4,3,2,1} & a_{7,5,3,2,1,0} & a_{7,3,2} \\ a_{7,3,2} & a_{7,6,2,1} & a_{6,4,2,0} & a_{5,3,1} & a_{4,2,0} & a_{7,5,4,3,2} & a_{6,4,3,2,1} & a_{4,3,0} \\ a_{4,3,0} & a_{7,3,2} & a_{7,5,3,1} & a_{6,4,2,0} & a_{5,3,1} & a_{6,5,4,3,0} & a_{7,5,4,3,2} & a_{5,4,1} \\ a_{5,4,1} & a_{4,3,0} & a_{6,4,2} & a_{7,5,3,1} & a_{6,4,2,0} & a_{7,6,5,4,1} & a_{6,5,4,3,0} & a_{6,5,2} \end{bmatrix}$$

and

$$m = [a_{6,0}, a_{7,6,1}, a_{7,2,0}, a_{6,3,1}, a_{7,6,4,2}, a_{7,5,3}, a_{6,4}, a_{7,5}]^T$$

Therefore, $P_{(M_{r,c}a'_{r,c}+m_{r,c})}$ is presented as

$$P_{(Ma'+m)} = a_0(a'_x + a'_y) + a_1 a'_x + a_2 a'_z + a_3 a'_4 + a_4(a'_y + a'_3) + a_5 a'_w + a_6(a'_z + \overline{a'_6}) + a_7(\overline{a'_5} + a'_4) = u'$$

(7)where

$$a'_w = a'_0 + a'_2 + a'_3 + a'_5, a'_x = a'_w + a'_7, a'_y = a'_1 + a'_4 + a'_5 \text{ and } a'_z = a'_2 + a'_7.$$

The 128-bit output of the SubBytes transformation acts as the input to ShiftRow, the output state of ShiftRows is obtained by just shifting the state entries in its input state. Hence the state entries in each row remain the same but differ by location.Therefore, by considering the output of ShiftRows and equation (5), for row r and column c, the output state of the flags can be rewritten as 16 formulations as follows:

$$e_{r,c} = P_{(M_{r,c^*}a'_{r,c^*}+m_{r,c^*})} + u'_{r,c^*}; \quad 0 \le r,c \le 3,$$

(8)

where c*= (r + c) mod 4. There by 16 error indication flags is generated from the output of ShiftRows for two transformations of SubBytes and ShiftRows together, i.e., one error indication flag for each byte, are obtained. This is shown in Fig.4.

4.1.2 MixColumns and AddRoundKey

The next two transformations in a typical AES encryption round are MixColumns and AddRoundKey. The MixColumns and AddRoundKey transformations are constructed matrix multiplication and modulo-2 addition of the input state with the roundkey respectively. Here low-complexity fault detection scheme derived for combined transformation of MixColumns and AddRoundKey.

Let $SR(A') = [a'_{r,c^*}]_{r,c=0}^{3}$ and $K - [k_{r,c}]_{r,c=0}^{3}$ be the input and the roundkey input of MixColumns and AddRound- Key in round i, respectively. Let the output of AddRoundKey be $O = [o_{r,c}]_{r,c=0}^{3}$ .Then, the following holds:

$$\sum_{r=0}^{3} o_{r,c} = \sum_{r=0}^{3} a'_{r,c*} + \sum_{r=0}^{3} k_{r,c}, \quad 0 \le c \le 3. \quad (9)$$

where c*= (r + c) mod . This can be rewritten as

$$\sum_{r=0}^{3}\left(a'_{r,c*} + k_{r,c} + o_{r,c}\right) = 0 \in GF(2^8), \quad 0 \le c \le 3 \quad (10)$$

and each summation is over GF (2^8) which consists of eight modulo-2 additions.

$$\sum_{r=0}^{3} a''_{r,c} = \sum_{r=0}^{3} a'_{r,c*}, \quad 0 \le c \le 3. \quad (11)$$

And we have

$$\sum_{r=0}^{3} o_{r,c} = \sum_{r=0}^{3} a''_{r,c} + \sum_{r=0}^{3} k_{r,c}, \quad 0 \le c \le 3 \quad (12)$$

Therefore,

$$\sum_{r=0}^{3} o_{r,c} = \sum_{r=0}^{3} a'_{r,c*} + \sum_{r=0}^{3} k_{r,c}, \quad 0 \le c \le 3 \quad (13)$$

Now let us introduce the four 8-bit error indication flags for four columns of the state as

$$E_c = \sum_{r=0}^{3}(a'_{r,c*} + k_{r,c} + o_{r,c}), \ 0 \le c \le 3 \quad (14)$$

In error-free situation, by using (10) all 32 bits of such flags in (14) must be zero, i.e. $E = 0 = (0, 0 \ldots 0) \in GF(2^3)$, $0 \le c \le 3$.

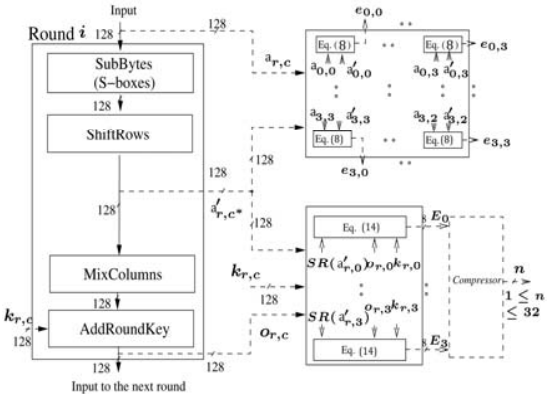

Figure.3 The proposed fault detection scheme for the i^{th} round of the AES encryption.

These 32 error indication flags can be used for these two combined transformations, i.e., eight error indication flags for each column of the state matrix as shown in Fig. 4.1. This error indication flags can be compressed so that n, $1 \le n \le 32$, error indication flags for these two transformations are achieved. This can be performed by ORing different combinations of the 32 error indication flags obtained in (14) as denoted by the compressor block in Fig.4. With 16 error indication flags by compression, greater than 99 percent of the errors are covered.

The last round of the every AES encryption (10^{th} round in AES-128 encryption) consists of all transformations, (SubBytes, ShiftRows, and AddRoundKey) except MixColumns transformation. Similar to all other rounds of the AES encryption, 16 error indication flags for SubBytes and ShiftRows combined can be used for the last encryption round. Consequently, (14) can also be used for the last round.

4.1.3 Further Enhancement

The complexity of the scheme is reduced by modifying the structure using subexpression sharing. This reduces the number of logic gates utilized in obtaining two sets of the error indication flags to have low-complexity fault detection scheme of the AES encryption, as shown in Fig.5. This is performed by modulo-2 addition of two sets of four coordinates of (14) for each column, i.e., $E_c = (e_{c,7}, e_{c,6}, \ldots, e_{c,0}) \in GF(2^8)$, $0 \le c \le 3$. Let $\hat{E}_c = (e_{c,4}, e_{c,2}, e_{c,1}, e_{c,0})$ and $\check{E}_c = (e_{c,5}, e_{c,7}, e_{c,6}, e_{c,3})$. Then, the four error indication flags of column c of the state are $\bar{E}_c = \hat{E}_c + \check{E}_c$; $0 \le c \le 3$. (15)

One can utilize four sets of modulo-2 additions of the output bits of each S-box pre-computed in (7), i.e., $a'_4 + a'_5, a'_2 + a'_7, a'_1 + a'_6$ and $a'_0 + a'_3$, to obtain the low-complexity error indication flags in (14). We use (14) to derive 16 low complexity signatures for the MixColumns and AddRound-Key transformations, i.e., four signatures for each column of the state matrix. This is shown in Fig. 5.This proposed fault detection for the MixColumns transformation which has 25 percent less areas overhead than the parity based scheme.

In fig 5, the Common Subexpressions (CSs) unit has been utilized to obtain 64 common subexpressions, i.e., 4 for each of the 16 S-boxes in the SubBytes transformation. If any of the two derived sets of error indication flags are one, the error is detected whereas if all of them are zero then no error has been detected although the output can be erroneous or correct. The (8) utilizes the hardware implementation of (7) which isless

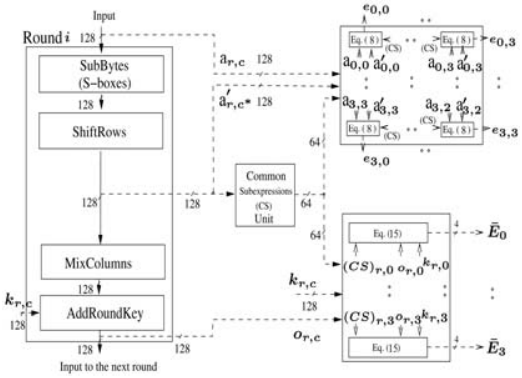

Figure.4 The low-complexity fault detection scheme utilizing subexpression sharing.

complex when the common sub expressions are used. Hence the Fig.5 shows less complexity compared to Fig.4.

4.2 AES Decryption

The AES decryption rounds also (except for the last round) consist of four transformations, i.e., InvShiftRows, InvSubBytes, AddRoundKey, and InvMixColumns. All the steps is similar to encryption but in reverse manner.

4.2.1 Inv Shift Rows and Inv Sub Bytes

In the AES decryption, the 128-bit input to InvShiftRows, i.e., the state matrix S' entries, is cyclically shifted to the right with the first row remaining unchanged.

$$e_{r,c} = P_{(M_{r,c}a'_{r,c^*}+m_{r,c})} + u'_{r,c}; \quad 0 \le r, c \le 3,$$

where c*= |r – c|. (16)

According to (15), 16 error indication flags for the Inv Shift Rows and Inv Sub Bytes transformations are generated.

4.2.2 ADDROUNDKEY AND INVMIXCOLUMNS

In decryption, InvMixColumns transformation is equivalent to multiplying the input state with the constant output matrix. In the AddRoundKey transformation, the input state, i.e., S, is added with the roundkey input state,i.e., K.

$$E_c = \sum_{r=0}^{3}\left(a_{r,c} + k_{r,c} + o_{r,c}\right), \quad o \le c \le 3$$

(17)

As in case of encryption, decryption also has three rounds in its last round i.e. InvMixColumns is removed. Similarly same (15) and (16) can be used in last round to detect fault.

For decryption also by using subexpression sharing the area overhead have been reduced 64 XOR gates to 48 XOR i.e. reduced by25 percent. Then, the four error indication flag for column c of state are

$$\bar{E}_c = \dot{E}_c + \hat{E}_c \ , 0 \le c \le 3 \qquad (18)$$

where $\dot{E}_c$ = ($e_{c,3}$, $e_{c,2}$, $e_{c,1}$, $e_{c,0}$) and $\hat{E}_c$= ($e_{c,7}$, $e_{c,6}$, $e_{c,5}$, $e_{c,4}$).

The proposed scheme has less area and critical path delay when compared to other schemes presented for InvMixColumns. It requires 48 XOR gates with two XOR in critical path delay. Overall 25 percent area overhead and 33 percent in critical path delay has been reduced in proposed scheme.

4.3 ERROR SIMULATION

When exactly 1 bit error appears at the output of the AES encryption or decryption rounds, the parity-based fault detection scheme is able to detect it and the error coverage will be 100 percent. But when there is case of multiple errors, the results of our simulations are valid.

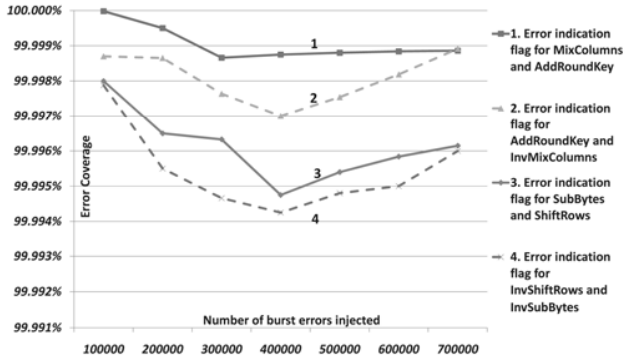

Figure.5 Simulation result for error coverage

We have considered both single and multiple stuck-at errors for the proposed scheme. These models cover both natural faults and fault attacks. In our simulations, we injected errors in two manners, i.e., burst and random errors, and obtain the error coverage for these two cases:

Burst Error:

In this type, the errors are injected at the 128-bit output of only one transformation in the AES encryption /decryption. The errors are monitored by injecting burst errors one at a time up to 700,000 at the transformation outputs. The error coverage for the two sets of error indication flags is greater than 99.996 percent.

Random errors:

This type of errors is injected at random locations, i.e., four 128-bit outputs of the transformations. The errors are covered either by one of the two series of the error indication flags.

The increase in the number of error injected increases the error coverage close to 100 percent. In Fig.7, the solid and dashed lines represent the error coverage for the AES encryption and decryption, respectively. For certain AES implementations containing storage elements, one can use the error correcting code-based approach presented in [13] in addition to the proposed scheme in this paper to make a more reliable AES implementation.

V. CONCLUSION

In this paper, we have considered a structure independent fault detection scheme for the AES encryption and decryption. This can be applied for both the S-boxes and the inverse S-boxes using lookup tables and those utilizing logic gates based on composite fields.using S-boxes and inverse S-boxes used for both LUT and composite fields. The proposed scheme has been simulated and its fault coverage has been evaluated in detail. The proposed system is able to find the round and its corresponding transformation in which fault occurred. Thereby optimized hardware is achieved bymodifying the structure using subexpresssion sharing. Hence the reduced number of gates is required in the implementation of AES.The slice overheads are less than those for the other schemes which have the same error coverage. Thus, this scheme has the highest efficiencies, showing reasonable area and time complexity overheads. Hence the proposed schemes outperform the previously reported ones.

VI. ACKOWNLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to all those who made the achievement possible without whom it would never have come into existence. I express my gratitude and thanks to my parents first for giving health as well as sound mind &financial support for taking up this paper. I would like to express my sincere appreciation and gratitude to Assit.Prof. M.Arul Kumar for his supervision and guidance during my studies and to develop this paper.

REFERENCES

[1] M. Mozaffari-Kermani and A. Reyhani-Masoleh, Member, “Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard,” IEEE Transactions on Computers, Vol. 59, No. 5, May 2010.

[2] C. Moratelli, F. Ghellar, E. Cota, and M. Lubaszewski, “A Fault-Tolerant DFA-Resistant AES Core,” Proc. IEEE Int’l Symp. Circuits and Systems (ISCAS ’08), pp. 244-247, May 2008.

[3] R. Karri, G. Kuznetsov, and M. Goessel, “Parity-Based Concurrent Error Detection of Substitution-Permutation Network BlockCiphers,” Proc. Int’l Workshop Cryptographic Hardware and Embedded Systems (CHES ’03), pp. 113-124, Sept. 2003.

[4] R. Karri, K. Wu, P. Mishra, and K. Yongkook, "Fault-Based Side Channel Cryptanalysis Tolerant Rijndael Symmetric Block Cipher Architecture," Proc. IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems (DFT '01), pp. 418-426, Oct. 2001.

[5] T.G. Malkin, F.X. Standaert, and M. Yung, "A Comparative Cost/Security Analysis of Fault Attack Countermeasures," Proc. Int'l Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC '06),pp. 159-172, Oct. 2006.

[6] C.H. Yen and B.F. Wu, "Simple Error Detection Methods for Hardware Implementation of Advanced Encryption Standard," IEEE Trans. Computers, vol. 55, no. 6, pp. 720-731, June 2006.

[7] M. Karpovsky, K.J. Kulikowski, and A. Taubin, "Differential Fault Analysis Attack Resistant Architectures for the Advanced Encryption Standard," Proc. Conf. Smart Card Research and Advanced Applications (CARDIS '04), vol. 153, pp. 177-192, Aug. 2004.

[8] X. Zhang and K.K. Parhi, "High-Speed VLSI Architectures for the AES Algorithm," IEEE Trans. Very Large Scale Integration Systems, vol. 12, no. 9, pp. 957-967, Sept. 2004.

[9] P. Maistri and R. Leveugle, "Double-Data-Rate Computation as a Countermeasure against Fault Analysis," IEEE Trans. Computers, vol. 57, no. 11, pp. 1528-1539, Nov. 2

[10] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Structure-Independent Approach for Fault Detection Hardware Implementations of the Advanced Encryption Standard," Proc. Int'l Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC '07), pp. 47-53, Sept. 2007.

[11] C. Moratelli, E. Cota, and M. Lubaszewski, "A Cryptography Core Tolerant to DFA Fault Attacks," Proc. Ann. Symp. Integrated Circuits and Systems Design (SBCCI '06), pp. 190-195, Sept. 2006.

[12] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "AParity Code Based Fault Detection for an Implementation of the Advanced Encryption Standard," Proc. IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems (DFT '02), pp. 51-59, Nov. 2002.

[13] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Lightweight Concurrent Fault Detection Scheme for the AES S-Boxes Using Normal Basis," Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '08), pp. 113-129, Aug. 2008.

BIOGRAPHY

S Anandi Reddy received the BE degree in Electronics and Communication Engineering in 2008, and ME degree in Communication System in 2011 from Dhanalakshmi Srinivasan Engineering College. Hermain interests include computer architecture and VLSI chip design.